COPPE/UFRJ

APLICANDO TILDECRF NA DETECÇÃO DE HOMOLOGIAS DISTANTES

Joel Bruno Santos da Costa

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientadores: Gerson Zaverucha
Vítor Manuel de Morais Santos Costa

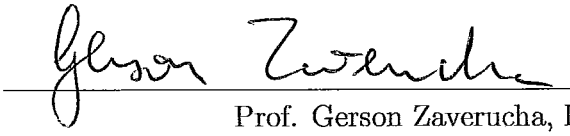
Rio de Janeiro
Setembro de 2008

APLICANDO TILDECRF NA DETECÇÃO DE HOMOLOGIAS DISTANTES

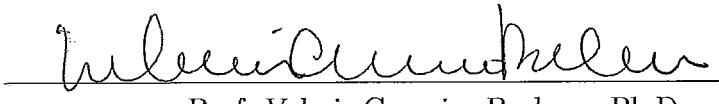
Joel Bruno Santos da Costa

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

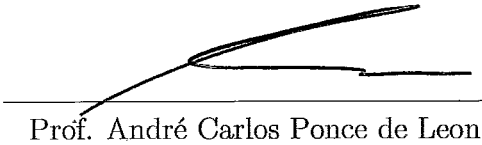
Aprovada por:



Prof. Gerson Zaverucha, Ph.D.



Prof. Valmir Carneiro Barbosa, Ph.D.



Prof. André Carlos Ponce de Leon Ferreira de Carvalho, Ph.D

RIO DE JANEIRO, RJ - BRASIL

SETEMBRO DE 2008

Costa, Joel Bruno Santos da

Aplicando TildeCRF na Detecção de Homologias Distantes/ Joel Bruno Santos da Costa. - Rio de Janeiro: UFRJ/COPPE, 2008

XIV, 76 p.: il.; 29,7 cm.

Orientadores: Gerson Zaverucha

Vítor Manuel de Morais Santos Costa

Dissertação (mestrado) - UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2008

Referências Bibliográficas: p. 64-76

1. Aprendizado de Máquina. 2. ILP. 3. Inferência Estatística. 4. Biologia Computacional. 5. Detecção de Homologias. I. Zaverucha, Gerson *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Dedicatória

Ao meu pai, meu exemplo de força de vontade

Agradecimentos

À minha querida mãe, Mercedes, por todo amor e carinho.

Ao meu querido pai, Aridio, por todo apoio e incentivo.

À minha querida esposa, Eliane, por não me deixar desistir e me ajudar a enfrentar todas as dificuldades.

A todos os familiares, por acreditarem.

Ao meu orientador, Professor Doutor Gerson Zaverucha, por me dedicar seu tempo e por confiar em mim para a elaboração deste trabalho.

Ao meu co-orientador, Professor Doutor Vítor Santos Costa, por responder pacientemente todas as minhas dúvidas e por estar sempre pronto a me ajudar.

A todos os amigos que fiz durante o mestrado pelo apoio, carinho e torcida, especialmente

a Juliana Bernardes, que deu início a este trabalho e me ajudou constantemente a dar continuidade a ele,

a Aline Paes e a Kate Revoredo, que estiveram sempre disponíveis e me ajudaram nas mais diversas tarefas,

e a Fábio Jimenez e a Luis Rigo, que me ajudaram com o cluster.

Obrigado a todos aqueles que torceram por mim.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

APLIACANDO TILDECRF NA DETECÇÃO DE HOMOLOGIAS DISTANTES

Joel Bruno Santos da Costa

Setembro/2008

Orientadores: Gerson Zaverucha
Vítor Manuel de Morais Santos Costa

Programa: Engenharia de Sistemas e Computação

Existe hoje em dia uma quantidade significativa de pesquisa envolvendo a detecção de homologias distantes entre seqüências de proteínas, um importante problema em Biologia Molecular Computacional. Os melhores resultados são obtidos com o uso de um método probabilístico denominado *profile hidden Markov models* (pHMM). Bernardes mostrou que a sensibilidade desses modelos aumenta significativamente ao adicionarmos informações estruturais no momento do treinamento.

Por outro lado, muitos trabalhos têm adotado modelos discriminativos, como os *Conditional Random Fields* (CRF), no lugar de generativos, como são os *hidden Markov models* (HMM), na solução de problemas de aprendizado de dados seqüencias. Em trabalho recente Gutmann e Kersting propuseram uma extensão de CRF, TildeCRF, onde as seqüências são formadas por átomos lógicos, incorporando toda a expressividade da lógica de primeira ordem às vantagens dos modelos discriminativos. Os resultados iniciais na predição de estruturas secundárias de proteínas foram promissores.

A principal contribuição desse trabalho foi desenvolver uma metodologia para usar o TildeCRF no problema de detecção de homologias distantes. Três tipos de experimentos foram feitos, cada um com um nível de informação diferente no treinamento. Os resultados foram comparados entre si, confirmando o aumento de acertividade também para o modelo discriminativo quando este dispõe de mais informações. Na comparação com programas específicos para o problema, a saber HMMER e HMMER-STRUCT, o TildeCRF não foi competitivo, mostrando que ainda muitos ajustes são necessários.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

REMOTE HOMOLOGY DETECTION APPLYING TILDECRF

Joel Bruno Santos da Costa

September/2008

Advisors: Gerson Zaverucha
Vítor Manuel de Morais Santos Costa

Program: Systems Engineering and Computer Science

Recently There has been a significant amount of research involving the detection of remote homologies between sequences of proteins, a major problem in Computational Molecular Biology. The best results are obtained using a method known as probabilistic *profile hidden Markov models* (pHMM). Bernardes showed that greater sensitivity can be achieved by using structural information when available.

In context, other fields of research have adopted discriminative models, such as *Conditional Random Field* (CRF), instead of generative ones, such as *hidden Markov model* (HMM), in the solution of problems of sequential data learning. In recent work Gutmann and Kersting proposed an extension of CRF, TildeCRF, where the sequences are formed by logical atoms, combining the expressiveness of the first order logic with the advantages of discriminative models. Initial results in predicting the secondary structures of proteins have been promising.

The main contribution of this work was to develop a methodology to use TildeCRF in the problem of detecting remote homologies. Three types of experiments were made, each with a different level of information in training. The results were compared with each other, confirming the increase of accuracy also for the discriminative model when it has more information. In comparison with specific programmes to the problem, namely HMMER and HMMER-STRUCT, the TildeCRF was not competitive, showing that many adjustments are still necessary.

Sumário

Lista de Abreviaturas

BLP *Bayesian Logic Programs*

CLP(BN) *Constraint Logic Programming*

CRF *Conditional Random Field*

FOLDT *First Order Logical Decision Tree*

glb *greatest lower bound*

HMM *Hidden Markov Model*

ICL *Independent Choice Logic*

ILP Programação em Lógica Indutiva

LOHMM *Logical Hidden Markov Model*

MGU *Most General Unifier*

pHMM *profile Hidden Markov Model*

PRM *Probabilistic Relational Models*

SLP *Stochastic Logic Program*

SLR *Statistical Relational Learning*

SCOP *Structural Classification of Proteins*

TDIDT *Top-down Induction of Decision Trees*

TILDE *Top-down Induction of Logical Decision Trees*

SVM *Support Vector Machine*

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Capítulo 1

Introdução

1.1 Motivação

Um importante problema em Biologia Molecular Computacional é a detecção de homólogos distantes, que são proteínas que têm um ancestral comum, mas divergem significativamente na sua história evolutiva. Uma série de ferramentas tem sido desenvolvida para este fim, tais como SAM (HUGHEY, KROGH, 1995) e HMMER (EDDY, 1998). Sem dúvida, uma abordagem eficaz e popular é a que traça o perfil de uma família de proteínas por meio de um modelo probabilístico chamado *profile hidden Markov model* (pHMM) (EDDY, 1998). Então, uma nova proteína é alinhada contra esses pHMM com o intuito de definir de qual família ela faz parte. Esses modelos são treinados, na maioria das vezes, apenas com informações da *estrutura primária* (seqüência de aminoácidos) das proteínas homólogas, apesar de trabalhos recentes demonstrarem que uma maior sensibilidade pode ser alcançada por meio da utilização de informações das estruturas secundária e terciária quando disponíveis (BERNARDES, 2005).

Hidden Markov model (RABINER, 1989) fornece um modelo generativo para dados seqüenciais (dados que podem ser dispostos em seqüências). Recentemente, tem havido um grande interesse em modelos discriminativos para seqüências de dados, como *Conditional Random Field* (CRF) (LAFFERTY, et al., 2001) e *Support Vector Machine* (SVM) (VAPNIK, 1995). Guntmann e Kersting apresentam o TildeCRF (GUTMANN, KERSTING, 2006), uma extensão para CRFs na qual

uma seqüência é formada por átomos lógicos, fornecendo assim uma plataforma de trabalho natural para expressarmos dados estruturados. Os resultados iniciais em testes de previsão de estrutura secundária de proteínas foram muito promissores.

1.1.1 O problema: detecção de homologia distante

Duas proteínas são homólogas caso tenham evoluído a partir de um ancestral comum e, provavelmente, compartilham a mesma função. Uma família de proteínas agrupa as proteínas homólogas. Então, dada uma nova proteína gostaríamos de avaliar se ela pode pertencer a uma determinada família, isto é, se ela é homóloga as proteínas de uma determinada família.

O método que vem sendo mais utilizado para detecção de homologias é a comparação por similaridade. O objetivo é encontrar sinais que as identifiquem como um possível membro do conjunto padronizado. Quando a similaridade entre as seqüências homólogas é baixa, elas ainda podem ser homóloga, neste caso homólogas distantes.

Uma forma de fazer a detecção é avaliar diretamente os alinhamentos de seqüências de aminoácidos que formam as proteínas, ou os alinhamentos estruturais das proteínas, de uma mesma família. Outra forma é criar modelos matemáticos a partir desses alinhamentos e obter assim um perfil da família.

1.1.2 Trabalhos relacionados

Abordagens tradicionais para detecção de homologias são baseadas em métodos que buscam por similaridade seqüencial (ALTSCHUL, et al., 1990) (PEARSON, 1990), ou seja, elas comparam a nova proteína com base de dados públicas, tais como GENBANK (BENSON, et al., 2005), TREMBL (BOECKMANN, et al., 2003) e SWISS-PROT (BAIROCH, BOECKMANN, 1993), buscando por similaridades seqüenciais entre a nova proteína e proteínas de função conhecida. Ou usam modelos probabilísticos, construídos a partir de alinhamentos múltiplos de seqüências homólogas, que descrevem o grau de conservação dos aminoácidos das proteínas (DURBIN, et al., 1998). Esses métodos funcionam bem para o reconhecimento de proteínas que possuem fortes similaridade em suas seqüências, mas falham na

detecção de *homologias distantes*. Duas proteínas são ditas homólogas distantes quando a similaridade entre suas seqüências é um fraco indicativo de homologia, mas a *estrutura tridimensional* (disposição espacial de seus átomos) é muitas vezes bem conservada, Bernardes (BERNARDES, 2005) mostra que alinhamentos obtidos por meio de informações estruturais podem levar a modelos mais expressivos do que os alinhamentos obtidos apenas das informações das seqüência.

Isso levanta a questão de como pode-se melhorar a detecção de homólogos distantes por meio do uso de informações estruturais. Uma abordagem baseia-se nas, amplamente disponíveis, implementações de pHMM, tais como SAM (HUGHEY, KROGH, 1995) e HMMER (EDDY, 1998). Alguns sistemas estendem o HMMER e consideram informações estruturais na construção dos modelos probabilísticos. Os bons resultados experimentais obtidos pelo HMMER-STRUCT (BERNARDES, 2005) mostraram que o uso de informações estruturais secundárias e terciária melhorou a qualidade do modelo.

HMMER e SAM são implementações muito eficientes de modelos generativos proposicionais para dados seqüenciais. Recentemente, muito trabalhos se dedicaram a investigar o uso de modelos discriminativos, tal como CRF (LAFFERTY, et al., 2001) e SVM (VAPNIK, 1995). Outra linha importante de trabalho investigou o uso de modelos lógicos para seqüências estruturadas. PRISM (SATO, KAMEYA, 1997), SLPs (MUGGLETON, 2002), CLP(BN) (COSTA, et al., 2008b), LoHMMs (KERSTING, et al., 2006), *Relational Markov networks* (RMN) (TASKAR, et al., 2002) , *Relational Markov models* (ANDERSON, et al., 2002) e TildeCRF (GUTMANN, KERSTING, 2006) são diferentes abordagens, mas todas podem ser usadas para modelar dados de seqüências estruturadas. PRISM e CLP(BN) fornecem uma plataforma de trabalho geral, que permite mais compactação na descrição dos modelos, e foram usados para implementar HMMs. Em contraste, LOHMM foi desenhado especificamente para lidar com seqüências de átomos lógicos. Uma alternativa bastante diferente foi incrementar Makov Fields para suportar lógica de primeira ordem, como fizeram os modelos de *Relational Markov networks* (RMN) e *Markov logic network* (MLN) (RICHARDSON, DOMINGOS, 2006). Seguindo o mesmo conceito dos LOHMMs, o TildeCRF pode ser visto como um passo na direção de restringir es-

sas ferramentas com grande expressividade para o propósito específico de lidar com seqüências lógicas.

CRFs foram aplicados com sucesso em várias aplicações de bioinformática tais como predição de estrutura secundária (GUTMANN, KERSTING, 2006), classificação de proteínas em *folds* (GUTMANN, KERSTING, 2006) (LIU, et al., 2005), alinhamento de seqüências de proteínas (DO, et al., 2006) e alinhamentos de estrutural de RNA (SATO, SAKAKIBARA, 2005).

1.2 Objetivos

O trabalho (BERNARDES, 2005) desenvolveu um novo método para treinar pHMMs considerando alinhamentos tridimensionais e diferentes propriedades estruturais sobre um conjunto de proteínas homólogas. Nesse método diferentes pesos são atribuídos a cada aminoácido do conjunto de proteínas alinhadas de acordo com a sua importância estrutural. Na abordagem seguida, as distribuições de probabilidades de emissão em pHMMs continuam sobre o alfabeto de aminoácidos, enquanto em abordagens anteriores, tais como (CHAKRABARTI, SOWDHAMINI, 2004), (GOYON, TUFFÉRY, 2004) e (BYSTROFF, BAKER, 2000), outros alfabetos especiais eram necessários.

Seguindo a mesma direção, nossa abordagem procurou manter o foco sobre o alfabeto de aminoácidos, mas, em vez de pesos, investimos na expressividade da lógica de predicados para adicionarmos as informações estruturais. Escolhemos como ferramenta o TildeCRF (GUTMANN, KERSTING, 2006) para os experimentos. Ele gera modelos discriminativos, usando CRFs, por meio da aprendizagem incremental de árvores de regressão. Esses modelos têm algumas vantagens sobre os modelos generativos, como pHMM.

Então, o objetivo principal deste trabalho é investigar se um método discriminativo de classificação de seqüências representadas por átomos lógicos pode obter bons resultados e ser competitivo na detecção de homólogos distantes.

1.3 Organização do trabalho

Os próximos capítulos dessa dissertação estão organizados da seguinte forma:

- o capítulo 2 define os conceitos biológicos que serão utilizados no decorrer do trabalho. Inicialmente, são descritas as proteínas e a formação de suas estruturas. Depois, é definido o conceito de homologia em proteínas e de forma isso é usado para agrupá-las;

- o capítulo 3 apresenta os modelos computacionais básicos relacionados a este trabalho. Os HMMs e seus algoritmos são abordados inicialmente e é apresentado uma aplicação específica para a classificação de proteínas, pHMM. Em seguida é feita uma comparação entre modelos generativos e discriminativos. Por fim, os CRF e os seus algoritmos são descritos;

- o capítulo 4 apresenta as ferramentas computacionais aplicadas em problemas de aprendizado relacional usadas nesse trabalho, em especial, modelos que usam lógica de primeira ordem. O Tilde (GUTMANN, KERSTING, 2006) e os conceitos lógicos de primeira ordem são discutidos no início do capítulo. Em seguida, é apresentada uma proposta de ferramenta que une HMM e lógica de primeira ordem LOHMM (KERSTING, et al., 2006). Seguindo a mesma linha de LOHMM, o TildeCRF une CRF e lógica de primeira ordem. Este é descrito no final do capítulo;

- o capítulo 5 apresenta a metodologia proposta para os experimentos, sendo definido a forma de conversão das seqüências de aminoácidos para a linguagem lógica de primeira ordem, assim como a incorporação de informações adicionais. Os resultados dos procedimentos de teste são descritos e os resultados são apresentados. Ao final, é realizada uma comparação do TildeCRF com o HMMER e HMMER-STRUCT;

- o capítulo 6 conclui o trabalho, apresentando uma discussão sobre os resultados alcançados e destacando possíveis caminhos para trabalhos futuros.

Capítulo 2

Conceitos Biológicos

Este capítulo apresenta resumidamente os conceitos básicos de biologia necessários para a compreensão desta dissertação. Similaridades estruturais e funcionais podem ser usadas para determinar a homologia entre duas proteínas. A seção 2.1 define proteína e suas estruturas. A seção 2.2 aborda o conceito de homologia e similaridade.

2.1 Proteínas

As proteínas são vitais para a vida celular (ALBERTS, et al., 2002), sendo responsáveis por muitas das funcionalidades dentro do organismos vivos. Por isso, é importante estudá-las.

Proteínas são compostos orgânicos complexos constituídos a partir de aminoácidos que são sintetizadas a partir de genes presentes no DNA. Para formar uma proteína os aminoácidos são dispostos em cadeias lineares que são estabelecidas por ligações denominadas ligações peptídicas. Uma proteína é um conjunto de cem ou mais aminoácidos, sendo os conjuntos menores, de até dez aminoácidos, denominados polipeptídeos.

Existem trezentos tipos de aminoácidos, porém somente vinte são encontrados no organismo humano , eles são chamados de aminoácidos primários ou padrões, apenas esses aminoácidos podem ser sintetizados pelo DNA humano. Veja a tabela 2.1.

As proteínas foram descritas pelo químico Jöns Jakob Berzelius, inicialmente, em 1838. Porém, somente em 1926, o importante papel das proteínas para os orga-

nismos vivos foi descoberto por James B. Sumner, quando ele mostrou que a enzima urease era uma proteína (SUMNER, 1926). Já a primeira proteína a ser seqüenciada foi a Insulina, por Frederick Sanger, em 1955.

Proteína	Símbolo	Abreviação
Glicina ou Glicocola	Gly, Gli	G
Alanina	Ala	A
Leucina	leu	L
Valina	Val	V
Isoleucina	Ile	I
Prolina	Pro	P
Fenilalanina	Phe, Fen	F
Serina	Ser	S
Treonina	Thr, The	T
Cisteina	Cis	C
Tirosina	Tyr, Tir	Y
Asparagina	Asn	N
Glutamina	Gln	Q
Aspartato ou Ácido aspártico	Asp	D
Glutamato ou Ácido glutâmico	Glu	E
Arginina	Arg	R
Lisina	Lys, Lis	K
Histidina	His	H
Triptofano	Trp, Tri	W
Metionina	Met	M

Tabela 2.1: Símbolo para os 20 aminoácidos primários.

Uma proteína pode ter quatro tipos de estrutura: primária, secundária, terciária e quaternária. Cada estrutura representa um estágio, veja a figura 2.1, da formação da estrutura final da proteína, que se inicia a partir da síntese da proteína.

A *estrutura primária* é a seqüência de aminoácidos ligados pelas pontes peptídicas como uma cadeia. Essa ligações possuem natureza planar e bastante rígida.

De acordo com interações intramoleculares, os aminoácidos mais próximos sofrem um arranjo espacial, formando a *estrutura secundária.* O arranjo secundário de uma proteína pode ocorrer de forma regular e os exemplos mais comuns são alfa-hélices (PAULING, et al., 1951) e folhas-beta (BRANDEN, TOOZE, 1991a). O arranjo também pode ocorrer de forma irregular sendo formado por regiões denominadas *coils* ou *loops* (BRANDEN, TOOZE, 1991a).

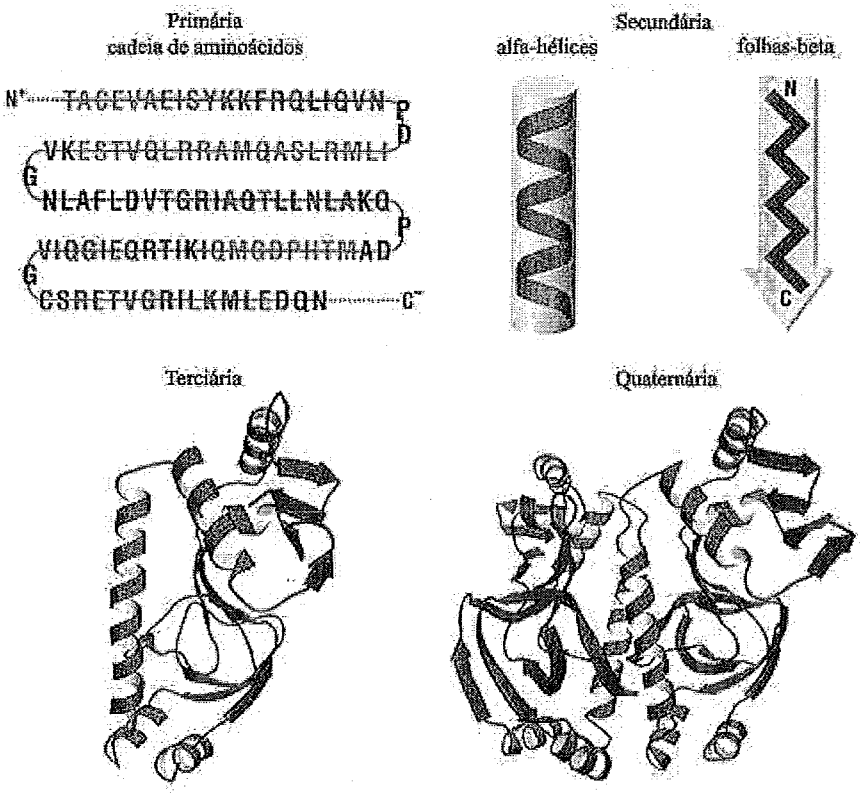


Figura 2.1: Fases da estrutura das proteínas (PETSKO, RINGE, 2004).

A *estrutura terciária* refere-se ao arranjo tridimensional de todos os átomos da proteína, completando a conformação da cadeia polipeptídica (BOURNE, WEISSIG, 2003). Ela descreve o dobramento final da cadeia sendo mantido por pontes de hidrogênio e dissulfídicas. Por um lado, a estrutura secundária apresenta interações moleculares de curta distância, por outro, a estrutura terciária apresenta as interações de longa distância e estas são entre estruturas secundárias.

Muitas dessas combinações de estruturas secundárias são freqüentemente encontradas em estruturas tridimensionais de diversas proteínas e algumas delas podem assumir papel funcional. Essas combinações são chamadas de *motivos* (BRANDEN, TOOZE, 1991b).

A maioria das proteínas é formada pela associação de várias estruturas terciárias, ou seja, várias cadeias polipeptídicas. Dessa união surge a *estrutura quaternária* que pode produzir diferentes funções para os compostos. Um exemplo de estrutura quaternária é a hemoglobina.

2.2 Homologia e Similaridade

Os organismos vivos compartilham propriedades bioquímicas, apesar das inúmeras diferenças entre eles (SCOTT, et al., 2000). Os processos evolutivos modificam as espécies dando-lhes características particulares e mantendo o que é importante dos seus ancestrais. Duas proteínas são homólogas caso tenham evoluído a partir de um ancestral comum.

Na bioinformática essa homologia é geralmente inferida com base em similaridade entre seqüências. Porém, similaridade não indica necessariamente que há um ancestral comum, logo nem sempre reflete homologia. Por outro lado, baixa similaridade não é suficiente para descartar a possibilidade de homologia. Nesta situação, os homólogos são chamados de homólogos distantes.

A seção 2.2.1 descreve os métodos para a detecção de seqüências homólogas, em especial, os programas de alinhamento, e a seção 2.2.2 define o SCOP, a base de dados que foi usada nos experimentos desta dissertação.

2.2.1 Alinhamento

As proteínas homólogas são geralmente divididas em famílias. Tais famílias servem de referência para a classificação de novas proteínas seqüenciadas em projetos genômicos, tanto pela comparação das seqüências, quanto pelo compartilhamento de funções. Os genes e proteínas seqüenciados passam por um processo conhecido como anotação, para que suas funções sejam definidas, após este processo os genes e proteínas são depositados em bancos de dados, tais como GENBANK (BENSON, et al., 2005), TREMBL (BOECKMANN, et al., 2003) e SWISS-PROT (BAIROCH, BOECKMANN, 1993). Posteriormente, essas bases de dados auxiliam o processo de anotação de novas seqüências.

A detecção semi-automática de seqüências homólogas reduz o tempo e o custo da classificação, economizando etapas nos testes laboratoriais, que têm um alto custo. Essa detecção é possível por meio de programas de alinhamento. BLAST (ALTSCHUL, et al., 1990) e FASTA (PEARSON, 1990) são exemplos de programas que alinham apenas pares de seqüências, isto é, são ferramentas de alinhamento par a par, ou seja, alinham uma proteína recém seqüenciada com cada proteína

de uma base de dados de proteínas anotadas. Já ferramentas como CLUSTALW (THOMPSON, et al., 1994), TCOFFEE (NOTREDAME, et al., 2000) e ALIGN-M (WALLE, et al., 2004) podem alinhar mais de duas seqüências e são conhecidas como ferramentas de alinhamento múltiplo.

A figura 2.2 mostra um exemplo de alinhamento de três seqüências de aminoácidos. Para alinhar as seqüências da melhor maneira possível, ou seja, tendo o máximo de resíduos idênticos alinhados, é necessário introduzir buracos no alinhamento das seqüências, esses são representados por um traço na figura 2.2. O termo resíduo, comumente, é utilizado para fazer referência a aminoácidos ou nucleotídeos. Os buracos são interpretados como inserções ou exclusões que ocorreram durante a evolução. Se duas seqüências de um alinhamento partilharem um ancestral comum, divergências ou lacunas poderão ser interpretadas como mutações pontuais provenientes do processo evolutivo. No alinhamento de proteínas, o grau de similaridade entre os aminoácidos que ocupam uma determinada posição pode ser interpretado como uma medida de conservação evolutiva.

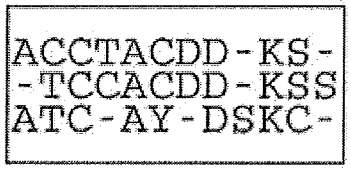


Figura 2.2: Alinhamento entre três seqüências de aminoácidos.

Uma grande variedade de algoritmos computacionais foram aplicados no problema de alinhamento de seqüências, incluindo programação dinâmica (BELLMAN, 1957) (SMITH, WATERMAN, 1981), métodos progressivos (THOMPSON, et al., 1994) (NOTREDAME, et al., 2000) e métodos probabilísticos (BATEMAN, et al., 2004) (HAFT, et al., 2003) (DO, et al., 2006).

A fim de melhorar a eficácia dos métodos a serem utilizados para a detecção de homologias distantes surge o critério de similaridade estrutural entre proteínas (CHAKRABARTI, SOWDHAMINI, 2004) (ESPADALER, et al., 2005) (ALEXANDROV, GERSTEIN, 2004) (HOU, et al., 2004). As estruturas tridimensionais das proteínas apresentam uma estrutura mínima, mesmo tendo sofrido alguma alteração quanto à organização de seus aminoácidos, que conserva as informações de sua

origem. Isso possibilita a comparação de proteínas quanto à origem. A figura 2.3 apresenta parte da estrutura tridimensional de duas proteínas homólogas distantes e a figura 2.4 mostra o alinhamento estrutural espelhado nas seqüências dessas mesmas duas proteínas. Podemos observar que as estruturas similares não correspondem necessariamente às regiões em que os aminoácidos são bem conservados.

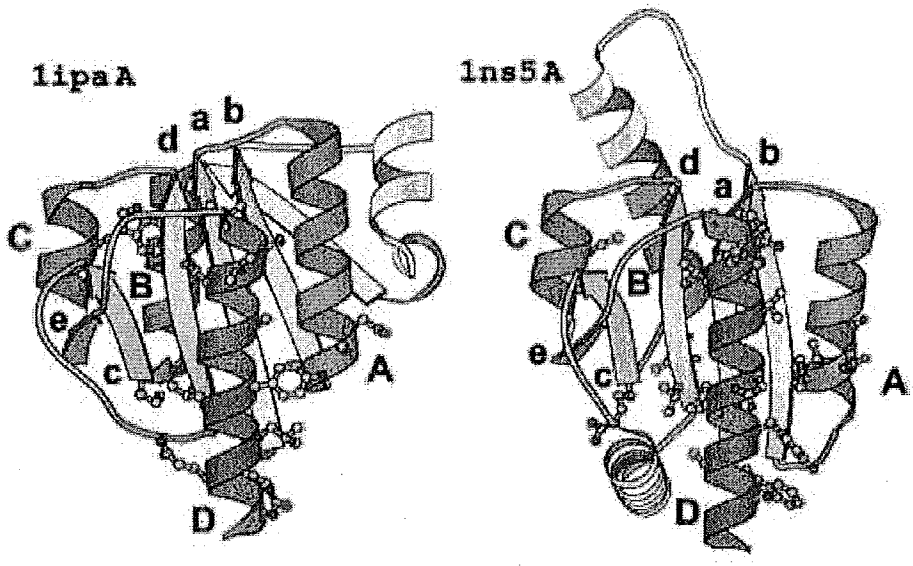


Figura 2.3: Estruturas de duas proteínas homólogas distantes (SADREYEV, GRISHIN, 2004). As estruturas secundárias similares são sinalizadas com letras maiúsculas (alfa-hélices) e minúsculas (folhas-beta).

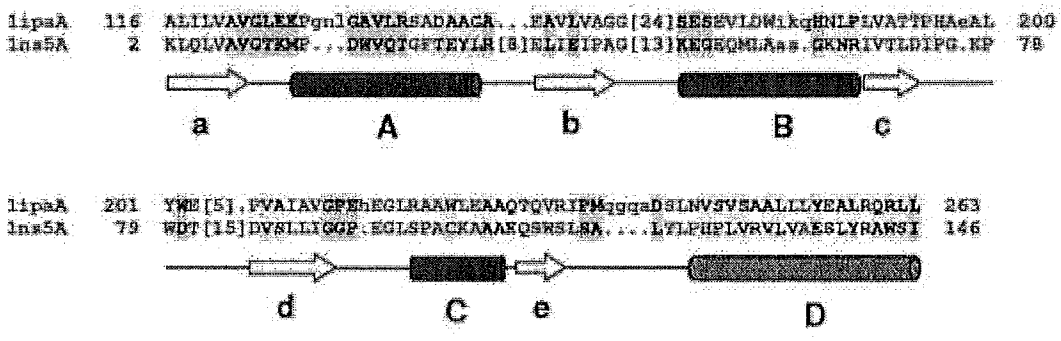


Figura 2.4: Alinhamento das seqüências das duas proteínas da figura 2.3 (SADREYEV, GRISHIN, 2004).

2.2.2 SCOP - Classificação Estrutural de Proteínas

Todas as proteínas de estrutura conhecida têm suas coordenadas espaciais inseridas na base PDB (BERMAN, et al., 2000) e, a partir dela, a base de dados

SCOP (ANDREEVA, et al., 2004) classifica todos os domínios dessas proteínas em uma ordem hierárquica com quatro níveis: família, super-família, dobramentos e classe.

No SCOP, uma *família* agrupa proteínas com identidade maior que 30% ou proteínas com funções e estruturas similares. Uma *super-família* é um conjunto de famílias em que as características estruturais sugerem um ancestral comum. As super-famílias são organizadas em *dobramentos* que possuem estruturas secundárias que compartilham ligações químicas e conexões topológicas. Por sua vez, os dobramentos são organizados em *classes* de acordo com a predominância dos elementos de estrutura secundária de suas proteínas. As classes são: *alfa*, em que as proteínas são essencialmente formadas por alfa-hélices, *beta*, onde as folhas beta são predominantes, *alfa e beta*, em que ocorrem muitas alfa-hélices e folhas beta intercaladas, *alfa mais beta*, com muitas ocorrências separadas de alfa-hélices e folhas beta e *múltiplos domínios*, no qual ocorrem diferentes dobramentos.

Muitos estudos usaram a base de dados SCOP para avaliar o desempenho de métodos voltados para a detecção de homologias, entre eles (ESPADALER, et al., 2005), (WISTRAND, SONNHAMMER, 2005), (SÖDING, 2005), (ALEXANDROV, GERSTEIN, 2004), (HOU, et al., 2004) e (BERNARDES, et al., 2007).

Capítulo 3

Modelos Computacionais

Este capítulo apresenta os métodos computacionais aplicados em problemas de classificação de proteínas usados neste trabalho. A seção 3.1 aborda os *hidden Markov models* (HMM), definindo os algoritmos mais utilizados de inferência e estimativa de parâmetros e, no final da seção, é apresentado uma aplicação específica para a classificação de proteínas. A seção 3.2 compara os modelos generativos com os discriminativos. A seção 3.3 define, a partir dos HMMs, os *conditional random fields* (CRF) e seus algoritmos de inferência e estimativa de parâmetros.

3.1 Hidden Markov Models

Inicialmente usados em aplicações de reconhecimento de voz (MENDEL, 1992), os *hidden Markov models* (RABINER, 1989) (HUGHEY, KROGH, 1996) são hoje em dia amplamente utilizados na análise de seqüências biológicas (MAMITSUKA, 2005) (EDGAR, SJÖLANDER, 2004) (KNUDSEN, MIYAMOTO, 2003) (BAE, et al., 2005) (CAMPROUX, TUFFÉRY, 2005) (LIN, et al., 2005) (BREJOVA, et al., 2005) (MAJOROS, et al., 2005).

O objetivo desta seção é descrever os conceitos básicos de HMM. Primeiramente, definiremos cadeia de Markov e HMM. E depois, as sessões 3.1.1 e 3.1.2 abordarão os algoritmos para inferência e estimativa de parâmetros, respectivamente. Profile HMMs (KROGH, et al., 1994) são usados para criar modelos de famílias de proteínas e serão apresentados na seção 3.1.3.

Geralmente *cadeias de Markov* são descritas como grafos direcionados, onde os nós são os estados e as arestas têm probabilidades associadas, que seguem a

propriedade de Markov, isto é, a probabilidade de estar em um estado só depende do estado imediatamente anterior, e não de toda a seqüência de estados trilhados até ali. Neste caso, dizemos que se trata de uma cadeia de primeira ordem. E, em uma cadeia de ordem k, o estado atual pode depender dos k estados anteriores.

As probabilidades associadas às arestas são portanto as probabilidades de transição entre os estados. Sejam dois estados s e t e uma aresta a que liga os dois, então a probabilidade de transição entre os estados s e t é $P(y_i = t|y_{i-1} = s)$. A soma das probabilidades de transição que saem de um estado deve ser igual a 1, $\sum_{\forall t} P(y_i = t|y_{i-1} = s) = 1$

A probabilidade de uma seqüência de estados é definida através do princípio da probabilidade condicional (DEGROOT, 1987). Em um modelo M, uma seqüência de estados y de tamanho n:

$$P(y, M) = P(y_n|y_{n-1})P(y_{n-1}|y_{n-2})...P(y_2|y_1)P(y_1) = P(y_1)\prod_{i=2}^{n} P(y_i|y_{i-1}) \quad (3.1)$$

A cadeia de Markov da figura 3.1 é um modelo observável, uma vez que, dada uma seqüência de observações qualquer, é possível afirmar a seqüência de estado visitados para emitir essa seqüência. Isso porque existe uma correspondência um para um entre estados e símbolos emitidos, isto é, cada estado emite apenas um símbolo.

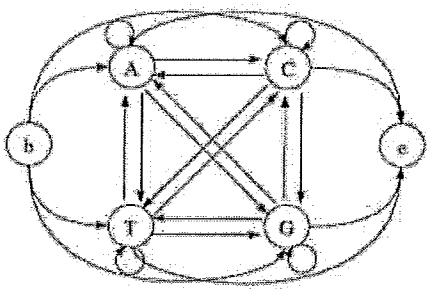


Figura 3.1: Cadeia de Markov para reconhecimento de seqüência de DNA, onde os estados A, C, T e G representam os nucleotídeos e b e e são estados de início e fim da cadeia.

Entretanto, se um estado emitir mais de um símbolo, é necessário uma função probabilística para as emissões neste estado. Suponhamos que todos os estados da figura 3.1 possam emitir os quatro símbolos usados na representação do DNA. Dada uma seqüência $x = TTGAC$, existem várias seqüências possíveis de estados

visitados. Dessa forma, a seqüência de estados que realmente emite os símbolos fica *escondida*.

Hidden Markov model (RABINER, 1989) é uma cadeia de Markov em que cada estado possui uma tabela de probabilidades de emissão para os símbolos possíveis do modelo. A soma das probabilidades desta tabela é igual a 1. Se existirem $b_1, ..., b_R$ símbolos que podem ser emitidos em um estado, então $\sum_{k=1}^{R} P(b_k) = 1$. A probabilidade conjunta de uma seqüência emitida x e uma seqüência de estados y de tamanho n é:

$$P(Y, X) = \prod_{i=1}^{n} P(y_i|y_{i-1})P(x_i|y_i) \tag{3.2}$$

A equação 3.2 especifica um HMM por meio de três distribuições de probabilidades. A primeira é a distribuição sobre os estados iniciais $P(y_1)$, igual a $P(y_1|y_0)$ para simplificar a notação. A segunda é a distribuição sobre as transições de estados, $P(y_i|y_{i-1})$. A terceira é a distribuição sobre as emissões, $P(x_i|y_i)$.

3.1.1 Inferência

Rabiner (RABINER, 1989) introduz a idéia que um HMM deve ser caracterizado por um dos três problemas a seguir:

1. dados um HMM $M(A, E)$, onde A é a matriz de probabilidades de transição e E é a matriz de probabilidades de emissão, e uma seqüência $X = X_1, X_2, ..., X_T$, computar eficientemente $P(X|M)$, isto é, a probabilidade da seqüência observada dado o modelo;

2. dados HMM $M(A, E)$ e X, determinar a melhor seqüência de estados escondidos $Q = q_1, q_2, ..., q_T$ para emitir X;

3. dados o HMM $M(A, E)$ e X, estimar A e E que maximizem $P(X|M)$.

Os dois primeiros são tratados nesta seção e o terceiro será visto na próxima.

Seja y um caminho em M composto por uma seqüência estados que inicia no estado inicial e termina no estado final. A verossimilhança (KARLIN, ALTSCHUL, 1990) da seqüência X descreve a probabilidade da seqüência ter sido gerada pelo modelo M através do caminho y. O primeiro problema descrito consiste em calcular

a verossimilhança de uma seqüência X dado o modelo M para todos os possíveis caminhos que gerem X, $\sum_y P(X|M)$. Por outro lado, a máxima verossimi-lhança indica o melhor caminho $y^* = \arg\max_y P(y|X, M)$, que é objetivo do segundo problema.

Como o número de caminhos pode crescer exponencialmente, os dois problemas podem ser computacionalmente muito difíceis. Porém, os métodos iterativos de programação dinâmica (BERTSEKAS, 1995) evitam a busca exaustiva de todos os caminhos possíveis.

Um HMM pode ser visto como um *grafo fator*, veja a figura 3.2, $p(y, x) = \prod_t \Psi_t(y_t, y_{t-1}, x_t)$ (SUTTON, MCCALLUM, 2006), onde $Z = 1$ e os fatores são definidos como:

$$\Psi_t(j, i, x) = p(y_t = j|y_{t-1} = i)p(x_t = x|y_t = j). \tag{3.3}$$

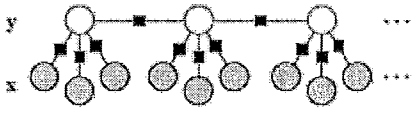


Figura 3.2: Exemplo de um HMM representado por um grafo fator.

O algoritmo *forward* (BALDI, BRUNAK, 2001) é um método que calcula a probabilidade da seqüência observada usando uma tabela para armazenar as probabilidades das sub-seqüências de X. A idéia por trás do algoritmo *forward* pode ser desvendada reescrevendo a soma $p(x) = \sum_y p(y, x)$ por meio da propriedade distributiva:

$$p(x) = \sum_y \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, x_t) \tag{3.4}$$

$$= (\sum_y \Psi_T(y_T, y_{T-1}, x_T))(\sum_y \Psi_{T-1}(y_{T-1}, y_{T-2}, x_{T-1}))...(\sum_y \Psi_1(y_1, y_0, x_1)) \tag{3.5}$$

Podemos observar que cada uma das somas intermediárias é usada repetidas vezes para computar a soma completa. Portanto, economiza-se trabalho de forma exponencial armazenando-se essas somas internas. Podemos definir, então, um conjunto de variáveis *forward* α_t, cada uma sendo um vetor de tamanho M, que é o número de estados, para armazenar essas somas intermediárias.

$$\alpha_t(j) = p(x_1, x_2, ..., x_t|y_t = j) = \sum_{y_{1...t-1}} \Psi_t(j, y_{t-1}, x_t) \prod_{t'=1}^{t-1} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}), \tag{3.6}$$

onde o primeiro somatório varia sobre todas as possíveis atribuições as variáveis aleatórias $y_1, y_2, ..., y_{t-1}$. Em outras palavras, $\alpha_t(j)$ é a probabilidade de a seqüência observada até x_t, inclusive, dado que o estado atual é j ($y_t = j$). Os valores de α podem ser computados pela equação recursiva

$$\alpha_t(j) = \sum_i \Psi_t(j, i, x_t)\alpha_{t-1}(i), \tag{3.7}$$

com inicialização $\alpha_1 = \Psi_1(j, y_0, x_1)$. É fácil verificar que $p(x) = \sum_j \alpha_T(j)$ através de repetidas substituições da recursão 3.7 para obter 3.6. O pseudo-código do algoritmo *foward* é apresentado no algoritmo 3.1.

Algoritmo 3.1 Forward

entrada: o Modelo M e uma observação X de tamanho T

$\alpha_0(0) = 1$
para todo estado j de M, $j > 0$ **faça**
  inicializa $\alpha_0(j) = 0$

para todo estado j de M, $j > 0$ **faça**
  para $t = 1, ..., T$ **faça**
    $\alpha_t(j) = \sum_i \Psi_t(j, i, x_t)\alpha_{t-1}(i)$

retorna matriz α

término $P(X|M) = \sum_j \alpha_T(j)$

O algoritmo *backward* (BALDI, BRUNAK, 2001) também é um método que calcula a probabilidade da seqüências observada usando uma tabela para armazenar as probabilidades das sub-seqüências de X, mas a recursão é feita do final para o início das seqüências. Veja o algoritmo 3.2. Portanto, temos a seguinte definição:

$$\beta_t(i) =_{def} p(x_{t+1}, x_{t+2}, ..., x_T|y_t = i) = \sum_{y_{t+1}...y_T} \prod_{t'=t+1}^{T} \Psi_{t'}(y_{t'}, y_{t'-1}, x_{t'}), \tag{3.8}$$

e a equação recursiva

$$\beta_t(i) = \sum_j \Psi_{t+1}(j, i, x_{t+1})\beta_{t+1}(j), \tag{3.9}$$

a qual é inicializada com $\beta_T = 1$. Assim, como no algoritmo *forward*, pode-se computar $p(x)$ por meio das variáveis *backward*, também usadas para armazenar somas intermediárias, $p(x) = \beta_0(y_0) = \sum_j \Psi_1(y_1, y_0, x_1)\beta_1(j)$.

Algoritmo 3.2 Backward

entrada: o Modelo M e uma observação X de tamanho T

para todo estado j de M, $j > 0$ **faça**
  inicializa $\beta_T(j) = 1$

para todo estado j de M, $j > 0$ **faça**
  para $t = T - 1, ..., 1$ **faça**
    $\beta_t(j) = \sum_i \Psi_{t+1}(j, i, x_{t+1})\beta_{t+1}(i)$

retorna matriz β

término $P(X|M) = \sum_j \Psi_1(y_1, y_0, x_1)\beta_1(j)$

Combinando os resultados das recursões do *forward* e do *backward* pode-se computar a distribuição marginal necessária na estimativa de parâmetros. Novamente, aplicando a regra distributiva

$$p(y_t = i, y_{t+1} = j|x) = \alpha_t(i)\Psi_{t+1}(j, i, x_{t+1})\beta_{t+1}(j). \tag{3.10}$$

O algoritmo *Viterbi* (BALDI, BRUNAK, 2001), veja o algoritmo 3.3, é um método para encontrar o caminho mais provável y^* que gerou uma determinada sub-seqüência, $y^* = \arg\max_y P(y|x)$. Pode-se observar que o artifício usado em 3.5 também funciona se substituirmos os somatórios por funções que retornam os máximos. Então, obtemos a recursão

$$\delta_t(j) = \max_i \Psi_t(j, i, x_t)\delta_{t-1}(i). \tag{3.11}$$

Vale observar que a variável *ptr* do algoritmo 3.3 armazena os ponteiros para os estados anteriores, então, a atual seqüência de estados pode ser encontrada por meio de *backtracking*.

3.1.2 Estimativa de parâmetros

O terceiro problema definido por (RABINER, 1989) é o mais difícil: determinar um método para ajustar os parâmetros do modelo $M(A, E)$ de forma a maximizar $P(X|M)$. Os parâmetros de um modelo HMM são as probabilidades de transição A e emissão E. A estimativa desses parâmetros é geralmente feita a partir de um conjunto de seqüências de exemplo, denominado conjunto de treinamento, por meio

Algoritmo 3.3 Viterbi

entrada: o Modelo M e uma observação X de tamanho T

$\delta_0(0) = 1$
para todo estado j de M, $j > 0$ **faça**
  inicializa $\delta_0(k) = 0$

para todo estado j de M, $j > 0$ **faça**
  para $t = 1, ..., T$ **faça**
    $\delta_t(j) = \max_i \Psi_t(j, i, x_t)\delta_{t-1}(i)$
    $ptr_t(j) = \arg\max_j(\Psi_t(j, i, x_t)\delta_{t-1}(i))$

retorna matriz δ e lista ptr

término $P(X, y^*|M) = \max_j(\delta_T(j))$
  $y_T^* = \arg\max_j(\delta_T(j))$

traceback
para $t = T, ..., 1$ **faça**
  $y_{t-1}^* = ptr_t(y_t^*)$

de algum procedimento iterativo, como, por exemplo, métodos EM (*Expectation-Maximization*) (DEMPSTER, et al., 1977), técnicas com gradiente (LEVINSON, et al., 1983) (BAGOS, et al., 2004) ou métodos de otimização numérica (COLLINS, 2002)

O principal método usado para estimativa de parâmetros é conhecido por *Baum-Welch* (BAUM, 1972), um caso especial de EM. Informalmente, os novos valores das probabilidades são calculados considerando os caminhos mais prováveis sobre as seqüências de treinamento e os atuais valores das probabilidades. Esse processo é repetido até que algum critério de parada seja atingido. É possível mostrar que a log-verossimilhança do modelo cresce a cada iteração, levando o processo a convergir para um máximo local.

Formalmente, o algoritmo *Baum-Welch* calcula, nos dados de treinamento, o número esperado de ocorrências da transição de um estado i para um estado j e o número esperado de ocorrências da emissão de um símbolo b a partir de um estado j. Sejam esses valores A_{ij} e $E_j(b)$. Para isso, são usados os mesmos valores obtidos pelos algoritmos *forward* e *backward* como mostrado na equação 3.10.

Então, é possível derivar o número esperado de vezes que a transição $i \rightarrow j$ é

usada por meio da soma em todas as posições e em todos as seqüências de treinamento,

$$A_{ij} = \sum_k \frac{1}{P(x^k)} \sum_t \alpha_t^k(i) \Psi_{t+1}(j, i, x_{t+1}^k) \beta_{t+1}^k(j), \tag{3.12}$$

onde $\alpha_t^k(i)$ é a variável *forward* $\alpha_t(i)$ definida em 3.7 e $\beta_{t+1}^k(j)$ corresponde a variável *backward* $\beta_{t+1}(j)$ definida em 3.9, ambas calculadas para a seqüência k. Da mesma forma, pode-se encontrar o número esperado de vezes que um símbolo b é emitido no estado j,

$$E_j(b) = \sum_k \frac{1}{P(x^k)} \sum_{t|x_t^k=b} \alpha_t^k(j) \beta_t^k(j), \tag{3.13}$$

onde a soma interna é somente sobre as posições t onde o símbolo emitido for b.

Uma vez calculados esses valores esperados, sendo a_{ij} a probabilidade de transição de i para j e $e_j(b)$ a probabilidade de emissão do símbolo b no estado j, então os novos parâmetros do modelo são calculados por meio de

$$a_{ij} = \frac{A_{ij}}{\sum_{j'} A_{ij'}} \qquad e \qquad e_j(b) = \frac{E_j(b)}{\sum_{b'} E_j(b')}. \tag{3.14}$$

Então, inicia-se uma nova iteração usando os novos valores dos parâmetros com o objetivo de estimar novos valores para as variáveis A_{ij} e $E_j(b)$ e, a partir deles, os novos parâmetros, seguindo assim, sucessivamente, até atingir algum critério de parada estabelecido previamente, por exemplo, se a mudança na log-verossimilhança for suficientemente pequena. É conveniente também determinar um número máximo de rodadas de forma a garantir a parada.

O pseudo-código do *Baum-Welch* é apresentado no algoritmo 3.4, no qual são usados pseudo-contadores para inicializar os contadores A_{ij} e $E_j(b)$. Esse procedimento é necessário, porque as equações 3.14 de atualização dos parâmetros são muito vulneráveis a ocorrência de *overfitting*, principalmente nos casos que existem dados insuficientes. De fato, se um estado k nunca é usado pelo conjunto de seqüências de exemplos, então as equações são indefinidas para este estado, ocorrendo zero tanto no numerador quanto no denominador. Existem alguns métodos para a atribuição de pseudo-contadores, entre eles o *zero-offset* (TATUSOV, et al., 1994), que adiciona uma pequena constante z a cada contador, *pseudo contadores* (KARPLUS, 1995), que generaliza o anterior usando um conjunto de constantes z_i diferentes, e *misturas*

Algoritmo 3.4 Baum-Welch

entrada: o Modelo M, um conjunto de n seqüências e um alfabeto de símbolos

inicializa M com valores arbitrários nos parâmetros

recursão
atribui pseudo-contadores às variáveis A_{ij} e $E_j(b)$, $\forall i, j, b$
para cada seqüência $k = 1, ..., n$ **faça**
  calcula $\alpha_t(i)$ para k através do algoritmo 3.1
  calcula $\beta_t(i)$ para k através do algoritmo 3.2
  adiciona a contribuição de k nas variáveis A_{ij} e $E_j(b)$ pelas equações 3.12 e 3.13

calcula os novos parâmetros do modelo usando as equações 3.14
calcula a nova log-verossimilhança do modelo

retorna o modelos atualizado e a log-verossimilhança

término pára ao atingir o critério de parada ou o número máximo de rodadas

de Dirichlet (SJÖLANDER, et al., 1996), que combina várias distribuições *Dirichlet*, que são distribuições multinomial (DEGROOT, 1987), para estimar a probabilidade na ausência de dados reais.

3.1.3 Profile HMM

Profile hidden Markov models (KROGH, et al., 1994)é um HMM bastante apropriado para modelar alinhamentos múltiplos de seqüências de gens ou proteínas. Mais especificamente, um *profile* HMM tem três tipos de estados: "*match*", "*delete*" e "*insert*". Um estado "*match*" emite um aminoácido com uma certa probabilidade de acordo com a sua posição. Um estado "*insert*" emite aminoácidos com uma certa probabilidade de acordo com uma distribuição já existente. E um estado "*delete*" é um estado que não emite símbolos correspondendo aos buracos nos alinhamentos múltiplos de seqüências (figura 2.2).

Assumamos que as seqüências da família que se deseja modelar estejam alinhadas. Para construir um modelo capaz de representar as propriedades de uma família é necessário identificar padrões na formação das seqüências envolvidas, como, por exemplo, consenso ou padrão de formação em determinadas colunas através da prevalência de algum aminoácidos ou ausência deles.

A seqüência consenso representa o ancestral comum, a partir do qual cada

proteína derivou. Sendo que algumas perderam aminoácidos durante a evolução, outras apenas substituíram o aminoácido original por outro e finalmente algumas proteínas ganharam novos aminoácidos. Porém, a seqüência consenso permanece caracterizando os membros da família. Para maiores detalhes sobre alinhamentos de seqüências consulte a seção 2.2.1.

Para representar os padrões das seqüências por meio de pHMM, é necessário representar as colunas do alinhamento que retêm os consensos da família. Se as colunas com buracos são desconsideradas, observa-se um conjunto de blocos bem formados, que unidos contêm a seqüência consenso capaz de representar os padrões da família.

A primeira arquitetura proposta para pHMM, apresentada na figura 3.3 (HENIKOFF, HENIKOFF, 1991), considera apenas blocos de seqüências alinhadas sem buracos, fazendo uso apenas de estados "match". Os resíduos anteriores e posteriores ao bloco são representados pelos estados N e S. Os estados b e e são silenciosos e marcam o início e fim do padrão representado. Essa arquitetura foi baseada nas *positions-specific scoring matrices* (GRIBSKOV, et al., 1987).

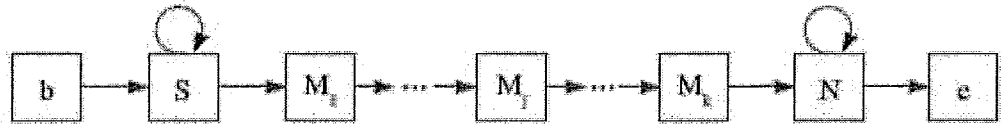


Figura 3.3: Primeira arquitetura de um pHMM.

Essa arquitetura evoluiu e passou a considerar os buracos por meio da inclusão de estados "inserts", I. Se tivermos dois blocos bem formados, os resíduos entre esses blocos serão as inserções. Essa arquitetura é adotada pelo programa META-MEME (GRUNDY, BAKER, 1997), que faz uso de pHMM para a detecção motivos. Veja sua representação gráfica na figura 3.4.

Porém, é necessário incluir o tratamento de exclusões com o uso dos estados "delete", D. Esses estados representam a ausência de resíduos em determinados estado de "match", ocasionando novas transições. Os estados D são silenciosos e seu propósito é adicionar saltos à arquitetura proporcionando uma melhor adaptação do pHMM aos padrões extraídos dos alinhamentos múltiplos de seqüências. A ar-

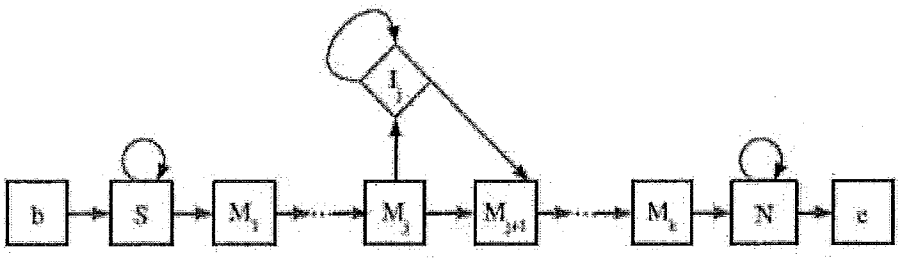


Figura 3.4: Arquitetura de um pHMM incluindo os estados de inserção.

quitetura completa, apresentada na figura 3.5, foi introduzida por (KROGH, et al., 1994). Esta arquitetura, com algumas alterações, é adotada pela maioria dos programas que implementam pHMMs na detecção de homologias, tais como HMMER (EDDY, 1998) e SAM (HUGHEY, KROGH, 1995).

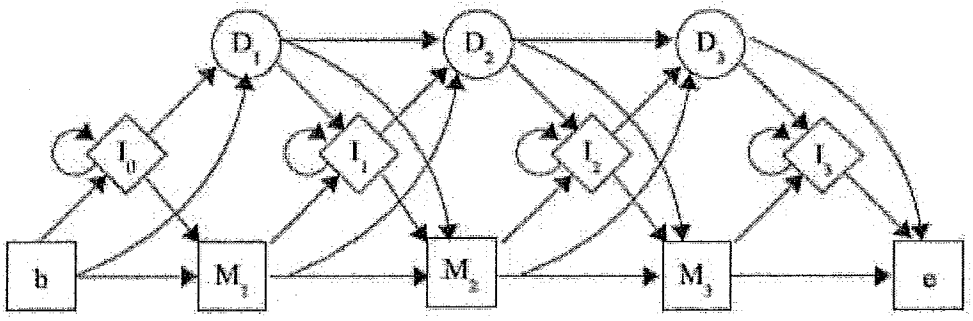


Figura 3.5: Arquitetura completa de um pHMM.

3.2 Modelos generativos e discriminativos

Os dois principais tipos de modelos nos problemas de aprendizado supervisionado são os *generativos* e os *discriminativos* (NG, JORDAN, 2002). Os modelos generativos modelam a distribuição conjunta $P(x, y)$, enquanto que os modelos discriminativos modelam a distribuição condicional $P(x|y)$.

A principal diferença entre eles é que a distribuição condicional não inclui um modelo de $p(x)$, o que não é mesmo necessário para a classificação. A dificuldade de modelar $p(x)$ é que ela geralmente contém muitas características altamente dependentes uma das outras. Há duas formas de incluir características interdependentes no modelo generativo: ou se amplia o modelo para representar estas dependências, ou se estabelece restrições de independência, assim como é feita no *naive* Bayes.

O sucesso dos modelos generativos se deve em grande parte as estas restrições assumidas nos modelos. Contudo, essas restrições nem sempre são verdadeiras. Em contraste, modelos discriminativos, como, por exemplo regressão logística (EFRON, 1975) e *suport vector machines* (VAPNIK, 1995), tipicamente fazem menos restrições sobre os dados e permitem que "os dados falem por si próprios". Foi demonstrado que em muitos domínios o modelo discriminativo é mais efetivo, como classificação de texto e mineração de dados, e existem alguns resultados empíricos mostrando que eles tendem a ter um erro assintótico mais baixo à medida que o tamanho do conjunto de treinamento aumenta (NG, JORDAN, 2002).

Lafferty *et al.* (LAFFERTY, et al., 2001) observaram que o fato de não precisarmos levar em consideração $p(x)$ pode explicar porque os *conditional random fields* tendem a ser mais robustos do que os modelos generativos ao violar as restrições de independência.

3.3 Conditional Random Fields

Os *conditional random fields* (LAFFERTY, et al., 2001) foram introduzidos como uma alternativa de modelo condicional no qual fosse possível relaxar as fortes suposições de independência feitas nos HMMs. Muito usados em processamento de linguagem natural (TASKAR, et al., 2002) (PENG, MCCALLUM, 2004) (SETTLES, 2005) (SHA, PEREIRA, 2003), os CRFs têm sido bastante aplicados em problemas de bioinformática, como por exemplo em alinhamento de seqüências de proteínas (DO, et al., 2006), alinhamento estrutural de proteínas (SATO, SAKAKIBARA, 2005) e classificação de proteínas em *folds* (LIU, et al., 2005).

Definiremos um caso particularmente importante de CRF, denominado *linear-chain* CRF (SUTTON, MCCALLUM, 2006) bastante apropriado para o modelo de seqüências. Para motivar nossa introdução, começaremos considerando a distribuição condicional $P(y|x)$ que segue a distribuição conjunta $p(y, x)$ de um HMM. O ponto chave que essa distribuição condicional é de fato a distribuição de um CRF com uma determinada escolha das *features*.

Primeiro, reescreveremos a equação 3.2 do HMM de uma forma que facilite a