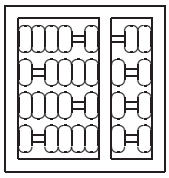

INSTITUTO DE COMPUTAÇÃO
UNIVERSIDADE ESTADUAL DE CAMPINAS

Visão Geral de um Sistema de Anotação de Proteínas de Transporte

Lin Tzy-Li and João Meidanis

Technical Report - IC-02-003 - Relatório Técnico

March - 2002 - Março

Visão Geral de um Sistema de Anotação de Proteínas de Transporte

Lin Tzy-Li* João Meidanis†
Inst. de Computação, Univ. Estadual de Campinas
Caixa Postal 6167, 13084-971 Campinas - SP
{lintzyli,meidanis}@ic.unicamp.br

Resumo

Proteínas de transporte são elementos importantes em todos os tipos de organismos. Em células humanas, doenças como fibrose cística e algumas formas de diabete são causadas por mutações em transportadores.

Uma comissão de especialistas em transportadores tem estudado estas proteínas há algum tempo e se encarregou em categorizá-las em famílias, dando-lhes um "TC number" (TC = Transport Commission). Assim, todo transportador deve ter um número TC.

Nos últimos anos foram disponibilizadas, no *site* do grupo de estudo do prof. Milton Saier, da UCSD [4], páginas que descrevem com detalhes estas famílias de transportadores e citam diversos exemplos de proteínas de transporte para cada família. Por serem muito informativas e possuírem milhares de proteínas-exemplo, o que permite automação e tratamento bioinformático, estas páginas foram uma das motivações do nosso projeto.

O nosso objetivo é desenvolver programas especialistas em reconhecer e classificar proteínas e sistemas de transporte, baseados na classificação TC.

Este documento apresenta a descrição de uma primeira versão das ferramentas, acompanhada de resultados preliminares e os planos para passos futuros.

1 Introdução

Proteínas de transporte são elementos importantes em todos os tipos de organismos. Em células humanas, doenças como fibrose cística e algumas formas de diabete são causadas por mutações em transportadores.

Uma comissão de especialistas em transportadores tem estudado estas proteínas há algum tempo e se encarregou em categorizá-las em famílias, dando-lhes um "TC number" (TC = Transport Commission). Assim, todo transportador deve ter um número TC.

*Financiada pelo Instituto Ludwig de Pesquisa sobre o Câncer-SP
†Financiado em parte pelo CNPq, FAPESP e Instituto Ludwig de Pesquisa sobre o Câncer-SP

Nos últimos anos foram disponibilizadas, no *site* do grupo de estudo do prof. Milton Saier, da UCSD [4], páginas que descrevem com detalhes estas famílias de transportadores e citam diversos exemplos de proteínas de transporte para cada família. Por serem muito informativas e possuírem milhares de proteínas-exemplo, o que permite automação e tratamento bioinformático, estas páginas foram uma das motivações do nosso projeto.

O nosso objetivo é desenvolver programas especialistas em reconhecer e classificar proteínas e sistemas de transporte, baseados na classificação TC.

Pretendemos refinar ao máximo os critérios de categorização, que podem ser específicos para cada família, para que possamos construir uma poderosa ferramenta de anotação automática de proteínas de transporte.

A ferramenta que pretendemos construir será formada por três módulos principais:

1. Nossa versão das páginas da classificação TC: será necessário construir uma versão local das páginas da classificação TC e uma ferramenta que faça a sincronização da nossa versão com a dele.

2. A ferramenta de anotação: dada uma seqüência (pedaço de genoma, ou gene, ou pedaço de gene), o programa deve determinar se esta seqüência é ou não é transportadora e, em caso afirmativo, classificá-la em uma ou mais famílias de transporte, reportando um número para cada categorização, que indica sua confiabilidade. Além disso, o programa deverá gerar um relatório com os critérios utilizados no processo de categorização, que deverá se basear principalmente na comparação da seqüência de entrada com os exemplos do site de classificação TC [4] (para esta comparação, pretendemos utilizar inicialmente PSI blast).

3. Benchmarks:

   - *Xylella*: Meidanis e outros [2] apresentam uma análise detalhada de todos os transportadores encontrado s no genoma da bactéria *Xylella fastidiosa*, usando a classificação TC, entre outros elementos. A lista destas proteínas, já classificadas com seus números TC, constitui este benchmark.
   - Exemplos de transporte: excluir cada exemplo de transporte do banco de proteínas construído a partir das páginas da classificação TC e testar o comportamento da nossa ferramenta ao tentar classificá-lo.

Na primeira parte, descrevemos as ferramentas que foram desenvolvidas inicialmente para testar e avaliar os resultados e os procedimentos necessários ao sistema de predição. Os programas desenvolvidos nesta fase são ativados em linha de comando. O objetivo final é que a ferramenta toda esteja disponível para uso via Web.

2 Sistema atual

O sistema atual é composto de vários programas executados separadamente, além de alguns procedimentos manuais para carregar tabelas no banco de dados (MySQL [3]) e consultá-las. Os programas são escritos em *Perl* e disparam outros programas usualmente utilizados no

dia-a-dia do laboratório de Bioinformática, como por exemplo BLAST e suas ferramentas auxiliares [1].

No sistema atual foram utilizados duas das várias versões e distribuições do BLAST, com a finalidade de compará-las e eleger um delas como padrão deste sistema: BLAST padrão e o PSI-BLAST. Porém, na maior parte deste documento, principalmente na parte descritiva, não iremos diferenciá-las, portanto BLAST designará esta família de ferramenta de busca de proteínas e seqüências similares de modo geral.

2.1 Processamentos

O fluxo de dados do processo atual é ilustrado pela Figura 1. Ele é usado tanto para calibrar como para predizer. A entrada do processo para ambos os casos são seqüências de proteínas. A saída, para o caso de calibragem, é a inclusão dos thresholds no banco de dados. No caso de predição, a saída é a avaliação da predição efetuada.

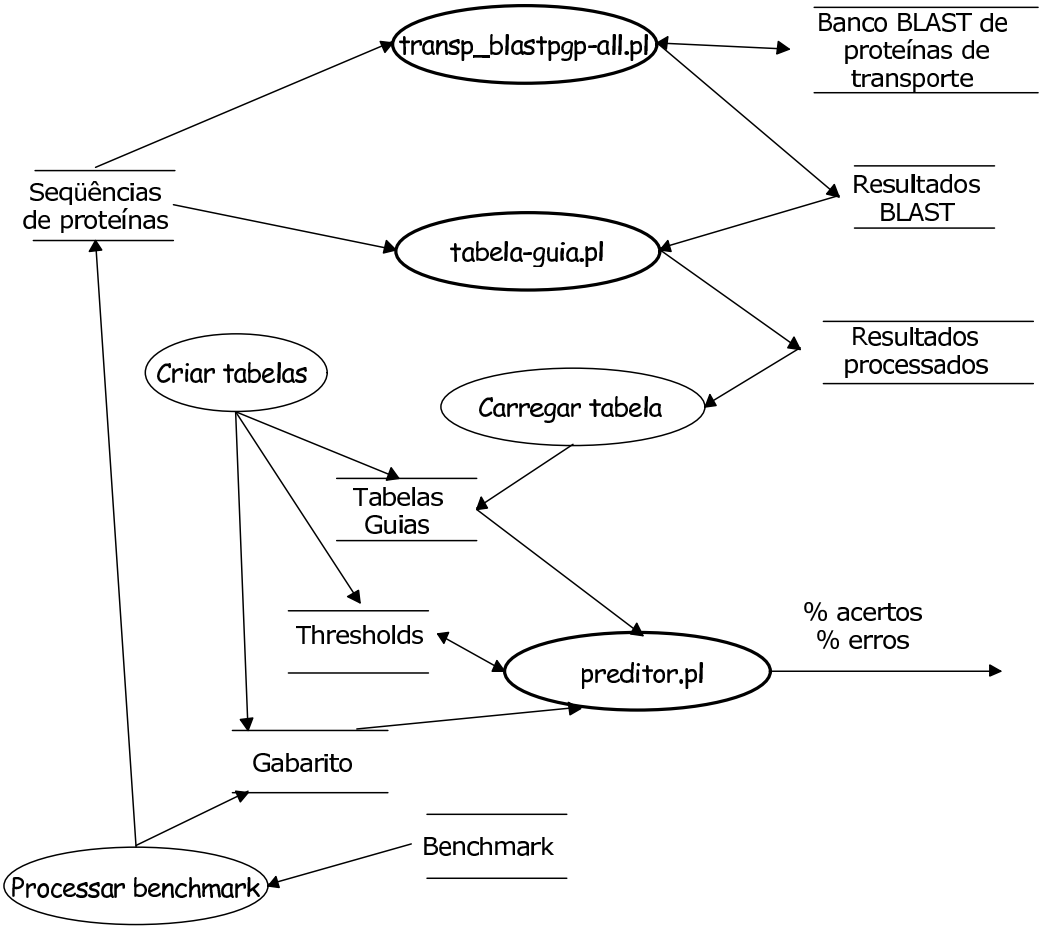

Figura 1: Visão geral do sistema atual.

2.1.1 Scripts

Os scripts envolvidos são:

transp_blastpgp-all.pl prepara o banco de proteínas de transporte — criando o banco de dados para o BLAST a partir de um arquivo FASTA com as seqüências das proteínas — e gerencia a execução do BLAST para cada uma das seqüências. Este módulo é usado tanto para calibrar como para predizer. Se a intenção é calibrar o banco de dados de proteínas, será feito o BLAST das seqüências de entrada contra o banco recém criado. Neste caso, o objetivo é verificar o comportamento do BLAST para as proteínas de várias famílias. Se o pretendido é apenas categorizar as seqüências de entrada, apenas será preciso executar o BLAST contra o banco de transporte previamente criado. As opções disponíveis pelo programa podem ser conferidas executando-o com a opção `-h`. Este script executa apenas o primeiro passo do processo de predição de proteínas de transporte.

tabela-guia.pl analisa os resultados do BLAST e extrai as informações obtidas para cada seqüência de entrada usada pelo programa anterior e produz arquivos para serem carregados no banco de dados MySQL. Estas informações serão utilizadas para calcular os parâmetros a serem usados pelos algoritmos de predição, no caso de calibragem, e para decidir se a proteína é ou não de transporte, em caso de a predição.

preditor.pl calcula os parâmetros (thresholds), por família, a serem usados pelos algoritmos, e avalia as predições feitas por cada algoritmo. A avaliação é feita com base no gabarito de alguma classificação previamente feita, que já deve estar no banco de dados como uma tabela. Quando do cálculo dos parâmetros, estes são inseridos diretamente no banco de dados.

Com exceção desses programas, todo o resto do processamento é feito manualmente.

2.1.2 Processamento Manual

As elipses com contornos mais grossos representam, na Figura 1, o processamento automático de informações, que neste caso específico são os programas descritos anteriormente. As com contorno mais fino ilustram processamentos manuais. Estes são descritos a seguir:

Carregar tabela consiste em um comando SQL que insere os registros contidos num arquivo texto em uma tabela do banco de dados.

Criar tabelas é um conjunto de procedimentos composto por: descrição em SQL da estrutura da tabela a ser criada e uso do comando SQL de criação para criar a estrutura descrita.

Processar benchmark é processar as informações do benchmark a ser usado, separando o gabarito das seqüências a serem classificadas pelo preditor de proteínas de transporte. Chamamos de benchmark o conjunto do arquivo FASTA com as seqüências das proteínas devidamente identificadas e da tabela em formato texto contendo uma

coluna com a identificação das seqüências e outra com o *TC number* da família em que ela deve ser classificada.

2.2 Dados armazenados

Os dados armazenados ou arquivos são representados, na Figura 1, por duas linhas paralelas envolvendo seu nome. Eles serão descritos a seguir e as tabelas referenciadas nesta seção estão todas no Apêndice A, página 15.

Banco BLAST de proteínas de transporte é um conjunto de arquivos que forma a base de dados do BLAST [1] para busca. Este arquivo geralmente é gerado por uma ferramenta do pacote de instalação do BLAST, que recebe como entrada um arquivo *FASTA*. O nome do banco usado atualmente é *transport-examples-new2*.

Seqüências de proteínas é um arquivo em formato *FASTA* das seqüências de proteínas a serem processadas. As seqüências podem ser todas de proteínas de transporte, neste caso podemos estar interessados em montar um banco de BLAST de proteínas, ou podem ser seqüências arbitrárias, situação em que queremos simplesmente identificar as proteínas de transporte do conjunto de entrada.

Resultados BLAST são arquivos gerados pelo BLAST, contendo os resultados da busca por similaridade de cada seqüência com o banco de proteínas do BLAST.

Resultados processados representa um arquivo texto, em que cada linha representa um registro de alguma tabela do tipo "Guia". Os campos do registro são separados por caracteres de tabulação.

Tabelas Guias são tabelas do banco de dados MySQL. Há duas versões para o layout destas tabelas. A versão para resultados provenientes do BLAST padrão (gapped) está descrita na Tabela 4 e a versão para resultados usando PSI-BLAST está descrita na Tabela 5, todas elas no Apêndice A. Os nomes usados correntemente são *Guia* e *GuiaPsi*, respectivamente.

Thresholds é a tabela de banco de dados que guarda os parâmetros a serem usados, dependendo da família predita para a seqüência, na predição dos resultados.

Esta tabela é particularmente interessante para os critérios de predição que se baseiam em resultados de BLAST. Os thresholds são parâmetros usados para determinar quando um *hit* deve ser considerado e analisado; por isso, são necessários parâmetros tanto para o BLAST padrão quanto para o PSI-BLAST, uma vez que eles funcionam diferentemente (ver seção 2.4). No sistema original há duas tabelas com a mesma estrutura, mas nomes diferentes: *ThresholdBlast* e *ThresholdPsi* respectivamente. A estrutura é descrita pela Tabela 6.

Gabarito é a tabela em banco de dados que armazena a família correta em que cada seqüência de proteína está classificada. Ela é usada para checar e avaliar outras predições feitas para estas mesmas seqüências. Sua estrutura com apenas dois campos

está na Tabela 7. Atualmente há apenas o gabarito da anotação de proteínas de transporte para a *Xylella fastidiosa* e o nome da tabela é *GabaritoXylella*.

Benchmark é um conjunto de arquivos composto por uma tabela em formato texto contendo uma coluna com a identificação das seqüências das proteínas e outra coluna com o *TC number* da família em que cada uma deve ser classificada, e um arquivo FASTA com as respectivas seqüências devidamente identificadas pelo nome que aparecem naquela tabela.

2.3 Métodos

A nossa estratégia de predição é baseada na análise dos resultados de BLAST. Os métodos de análise foram testados tanto para os resultados do BLAST padrão, como para os do PSI-BLAST. A diferença é que o segundo vai além da iteração produzida pelo BLAST padrão, refinando o resultado em outras iterações. Maiores detalhes sobre os dois podem ser obtidos na Seção 2.4. A saída gerada pelo PSI-BLAST é representada parcialmente pela Figura 2.

Há três métodos idealizados para fazer a predição de proteína de transporte que classificam a seqüência de entrada como tal:

A. usar apenas o melhor *hit*,

B. usar os N primeiros *hits*,

C. usar os *hits* com pontuação (score) maior ou igual a S.

2.3.1 Parâmetros N e S

O N e o S são parâmetros pré-definidos para cada família e determinados de acordo com o conjunto de exemplos de proteínas de transporte e o tipo de BLAST usado para garimpar os dados.

Esses parâmetros são obtidos analisando-se os resultados do processamento com BLAST das próprias seqüências de proteínas que compõem o banco de dados do software contra elas mesmas, que chamaremos de seqüências exemplos.

Para cada seqüencia-exemplo é contabilizado quantos dos seus melhores hits no BLAST são da mesma família. Este número será denominado **Ntops**. O N é o *Ntops* que aparece com mais freqüência para uma determinada família.

Já o S, denominado **Score**, possui duas fórmulas de cálculo. As variáveis da fórmula podem ser melhor compreendida observando-se também a Figura 2. Tudo depende se a família é boa ou não. Uma família, f, é boa se todas as seqüências-exemplo da família satisfizerem a condição:

$$Max(BestOutFamilyScore) <= Min(BestHitScore),$$

sendo $BestHitScore$ a pontuação (score) do melhor *hit* do BLAST, o que normalmente é o próprio exemplo, e o $BestOutFamilyScore$ é o score do melhor *hit* fora da família exemplo.

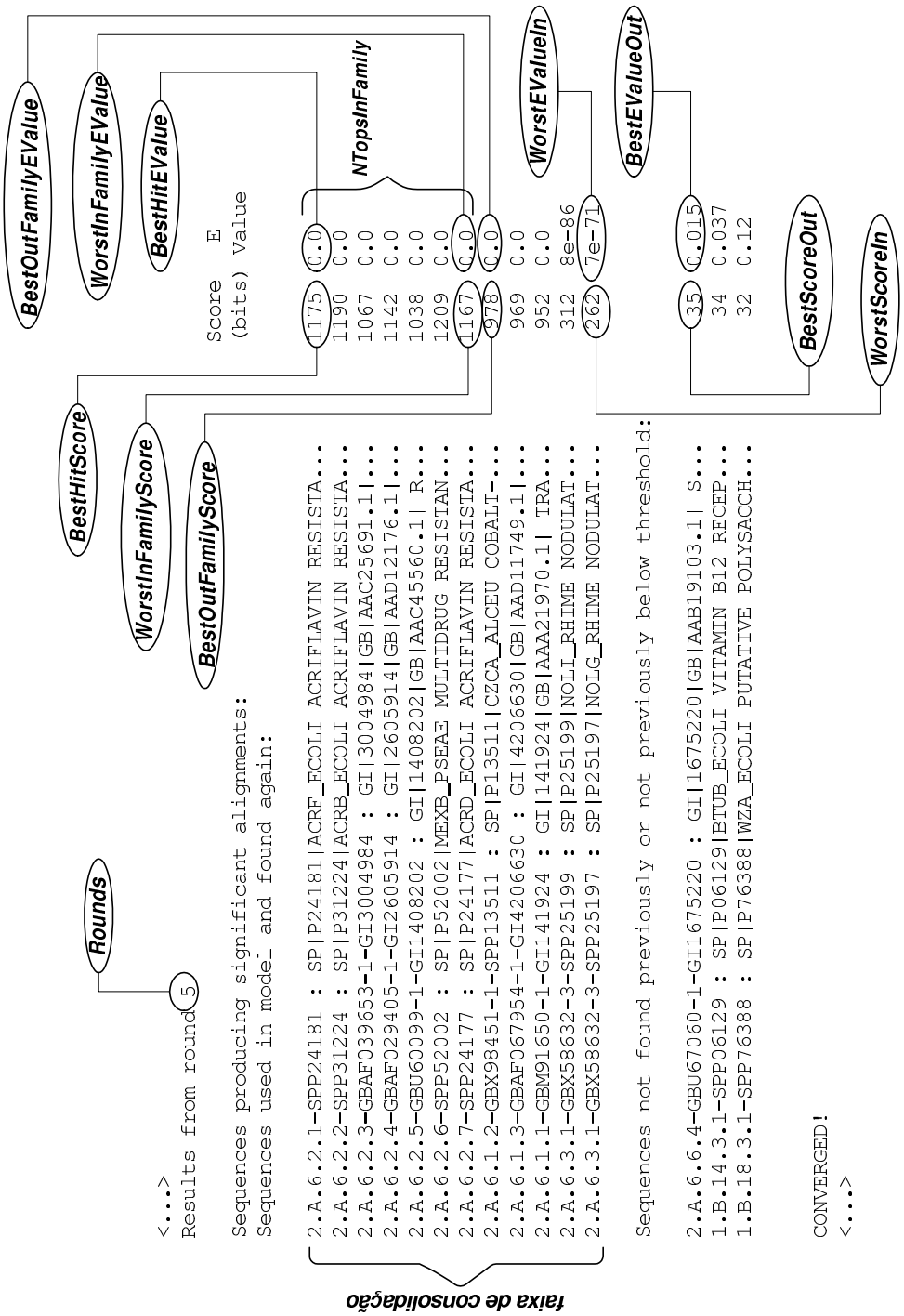

Figura 2: Saída (parcial e editada) do Psi-BLAST. Explicativo sobre alguns campos da tabela GuiaPsi.

Para as famílias boas, a fórmula usada é:

$$S(f) = [max(BestOutFamilyScore) + min(BestHitScore)]/2,$$

enquanto para as outras famílias a fórmula é:

$$S(f) = min(BestHitScore).$$

2.4 BLAST padrão versus PSI-BLAST

O BLAST padrão, conhecido como *Gapped BLAST* [1], é bastante popular na comunidade científica, pois é rápido para procurar a similaridade de uma seqüência com outras do seu banco de dados. A cada seqüência do banco reportada pelo BLAST como sendo similar ao da entrada é usualmente conhecida como ***hit***.

Há várias versões da ferramenta BLAST. Cada uma delas com melhorias específicas para atender melhor um determinado caso que outros. O PSI-BLAST age em várias iterações e a iteração anterior influencia o resultado do passo seguinte, sempre refinando a pontuação dada para a similaridade com a seqüência encontrada. O número máximo de iterações pode ser definido pelo usuário, porém, se acontecer algum dos casos a seguir, as iterações podem parar antes delas atingirem a quantidade definida pelo usuário:

- o resultado da iteração atual é igual ao da iteração anterior,
- o refinamento da iteração excluiu todos os resultados encontrados anteriormente,
- o BLAST não encontrou nenhuma seqüência em seu banco de dados que fosse parecida com o resultado.

2.4.1 Benchmarks

O desempenho dos métodos analisando os resultados da comparação do próprio banco de proteínas de transporte contra ele mesmo, usando o BLAST e o PSI-BLAST, é apresentado em percentual de acertos e erros na Tabela 1. Já a Tabela 2 mostra os resultados dos métodos usando os resultados das duas versões do BLAST das proteínas da *Xylella fastidiosa* contra o banco de proteínas de transporte. Do conjunto de seqüências da *Xylella*, foram excluídas as proteínas sem classificação definida, desconhecidas, ou hipotéticas.

Método	**BLAST**		**PSI-BLAST**	
	Acerto	Erro	Acerto	Erro
A	99.2	0.8	94.2	5.8
B	81.8	18.2	82.6	17.4
C	95.3	4.7	76.8	23.2

Tabela 1: Percentual de acerto dos métodos: banco BLAST contra ele mesmo (total: 1726 proteínas).

Pelas análises com seqüências exemplos com proteínas de *Xylella fastidiosa*, os resultados com maior números de acertos na predição eram os provenientes do PSI-BLAST, portanto este foi o escolhido para ser usado como padrão.

Método	**BLAST**		**PSI-BLAST**	
	Acerto	Erro	Acerto	Erro
A	14.3	85.7	88.9	11.1
B	62.9	37.1	90.9	9.1
C	86.5	13.5	89.2	10.8

Tabela 2: Percentual de acerto dos métodos: banco BLAST contra proteínas de *Xylella fastidiosa* (total 1283 proteínas).

3 Nova Proposta

Os resultados obtidos com as implementações preliminares nos fizeram repensar a ferramenta de predição como descrito a seguir.

Idealmente a ferramenta pode ser dividida nos seguintes módulos:

Calibrador recebe um arquivo no formato *FASTA* contendo exemplos de seqüências de proteínas de transporte que servirá como base para definir os parâmetros usados nos algoritmos de predição.

Preditor recebe um arquivo em formato *FASTA* com as seqüências a serem preditas e retorna a família prevista para cada seqüência de entrada, ou uma mensagem dizendo que não é proteína de transporte. O ideal seria que a previsão também viesse acompanhada de um grau de confiabilidade.

Avaliador recebe um arquivo em formato *FASTA* com seqüências acompanhada de uma classificação (*benchmark*, ver definição na página 6), retira a classificação, passamos pelo preditor, e estima os erros e acertos da predição. As estimativas deverão ser baseadas num grau de confiabilidade.

A Figura 3 mostra como a integração entre os módulos seria feita.

3.1 Projeto do banco de proteínas de Transporte

A Figura 4 traz a modelagem do que será o banco de dados de proteínas de transporte. Ela já detalha a décima versão do diagrama entidade-relacionamento.

As escolhas de projeto das entidades e seus relacionamentos são vinculadas à maneira como as páginas da classificação TC [4] apresentam as famílias das proteínas de transporte e os exemplos que estão associadas a cada uma delas. Como a página é em inglês, o conteúdo do banco de dados também será em língua inglesa.

Um ponto controverso é a possibilidade de uma proteína pertencer a mais de uma família. Nas páginas da classificação TC [4], versão do dia 6 de maio de 2000, havia vários casos assim. As famílias em que isso aconteceu estão listados a seguir e a Tabela 3 mostra a relação entre elas. Na coluna "Família" aparece o número *TC* das famílias aos quais as proteínas, que aparece na coluna "Intersecção", pertencem.

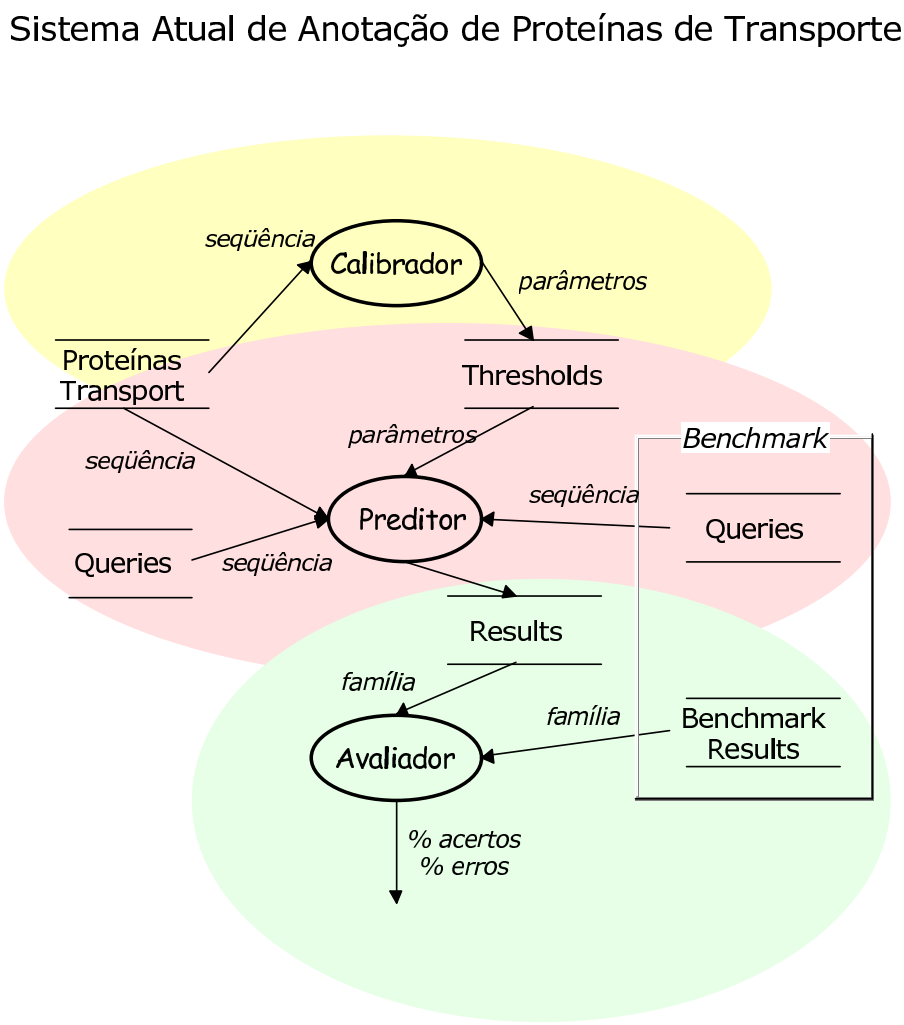

Figura 3: Visão geral da nova formulação.

1.B.22 The Outer Bacterial Membrane Secretin (Secretin) Family
2.A.5 The Zinc (Zn2+)-Iron (Fe2+) Permease (ZIP) Family
2.A.17 The Proton-dependent Oligopeptide Transporter (POT) Family
2.A.45 The Arsenite-Antimonite (ArsB) Efflux Family
2.A.74 The 4 TMS Multidrug Endosomal Transporter (MET) Family
3.A.5 The Type II (General) Secretory Pathway (IISP) Family
3.A.6 The Type III (Virulence-related) Secretory Pathway (IIISP) Family
3.D.3 The Proton-translocating Quinol: Cytochrome c Reductase (QCR) Superfamily
3.E.2 The Photosynthetic Reaction Center (PRC) Family
9.B.12 The (Salt or Low Temperature) Stress-induced Hydrophobic Peptide (SHP) Family
9.B.18 The SecDF-associated Single Transmembrane Protein (SSTP) Family
9.B.19 The Mn2+ Homeostasis Protein (MnHP) Family
9.B.28 The YqaE Family

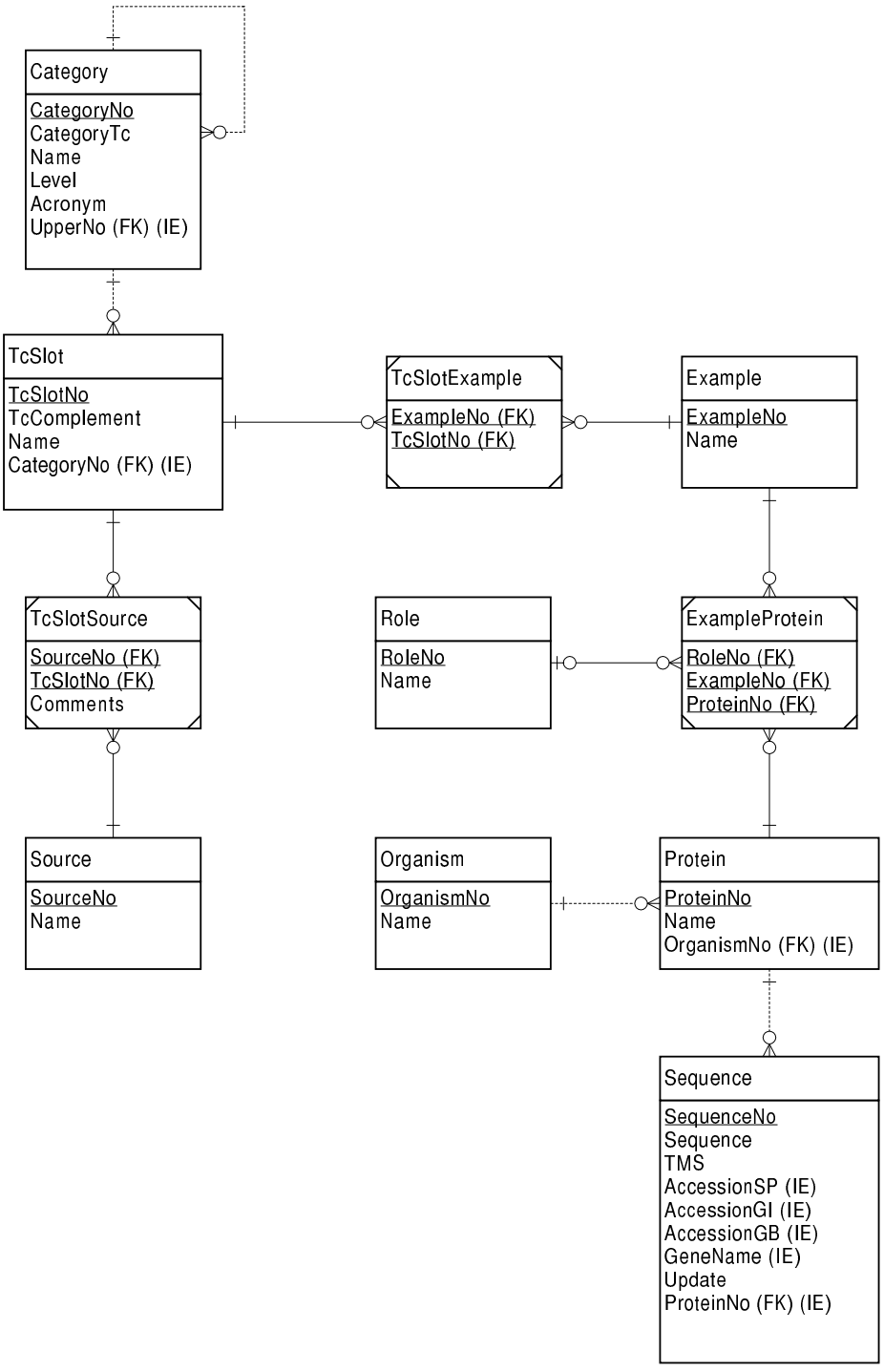

Figura 4: DER do banco de dados de proteínas de Transporte.

Famílias		**Intersecção**
1.B.22	3.A.5	SPP15644 - PulD protein secretin SPP35818 - XcpQ secretin protein
1.B.22	3.A.6	SPQ01244 - YscC of *Yersinia enterocolitica*
2.A.17	2.A.45	SPP36574 - DtpT of *Lactococcus lactis*
2.A.5	9.B.19	SPQ12067 - ATX2 of *Saccharomyces cerevisiae*
2.A.74	9.B.6	SPQ60961 - MTP of *Mus musculus* SPQ61168 - E3 protein of *Mus musculus*
3.A.5	9.B.18	SPP19677 - YajC (subsistema de SecAYEGDF of *E. coli*)
3.D.3	3.E.2	SPP26287 - Cytochrome f protein SPP26290 - Fe2S2 protein
9.B.12	9.B.28	SPP77240 - YgaE of *E. coli* SPQ42509 - BLT101 of *Lophopyrum elongatum*

Tabela 3: Famílias com proteínas exemplos em comum.

Também pode parecer estranho o fato da modelagem permitir que uma proteína possua mais de uma seqüência. No entanto, há proteínas, por exemplo a hemoglobina, que é composta por mais de um gene e só faz sentido falar em seqüências desta proteína, quando levado em consideração todas as seqüências dos genes que as compõem; por exemplo, a família *1.A.6 – The Epithelial Na+ Channel (ENaC) Family* – possui vários ítens que se enquadram nessa situação. Já no caso de se tratar de um sistema, as proteínas que o compõem têm existência e função própria, por isso elas são denominadas subunidades. Um exemplo de sistema são os ítens da família *3.A.1.3 – The Polar Amino Acid Uptake Transporter (PAAT) Family.*

Outra decisão de projeto: como cada parte do número TC identifica um grupo, subgrupo e família ao qual uma proteína de transporte pertence, isso nos levou à criação de duas entidades a partir de um número TC: "Category" e "TcSlot". O exemplo a seguir servirá para explicar estas entidades.

No número TC *1.A.35.1.1*, temos cinco casas separadas por pontos (.), que chamaremos de *slot*. O primeiro define a classe e é também o primeiro nível da categoria, que conjugado com o segundo slot informa a super-família do item e identifica o segundo nível ao qual o item pertence e assim por diante (Figura 5). O terceiro nível identifica a família, no entanto há casos em que a super-família está no terceiro nível e as famílias no quarto nível. Este é o caso, por exemplo, de *2.A.1* (The Major Facilitator Superfamily (MFS)).

3.2 Banco de benchmarks

Como estaremos trabalhando com avaliação da predição, é necessário um banco para gerenciamento dos benchmarks, ou seja, dos conjuntos de seqüências previamente classificadas por algum outro meio confiável qualquer. Espera-se que a classificação do preditor coincida com a do benchmark. Na Figura 6 temos um diagrama esboçando o que seria o banco de benchmarks.

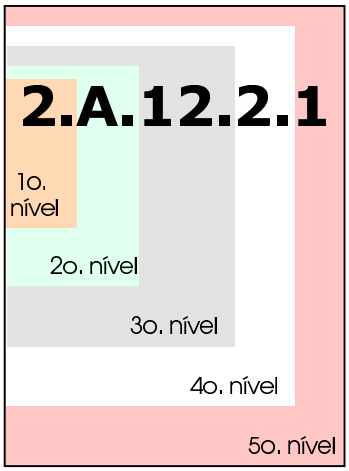

Figura 5: Um exemplo dos níveis de um número TC.

4 Conclusões

Este documento descreve a implementação inicial de um sistema de predição de proteínas de transporte. Além disso, uma nova proposta para melhor modularização do sistema, com mais funcionalidade e mais flexibilidade, é também descrita.

Estão em preparação documentos adicionais descrevendo em mais detalhes cada um dos módulos identificados na nova proposta.

5 Agradecimentos

À Marília D. V. Braga [2] pelo fornecimento do arquivo FASTA, devidamente "curado", das seqüências-exemplos de proteínas de transporte extraídas do site da classificação TC [4], e pelas contribuições para o design do novo banco de dados de proteína de transportes descrito na seção 3.1.

Ao Instituto Ludwig de Pesquisa sobre o Câncer, à Fapesp e ao CNPq pelo financiamento das bolsas dos envolvidos neste projeto.

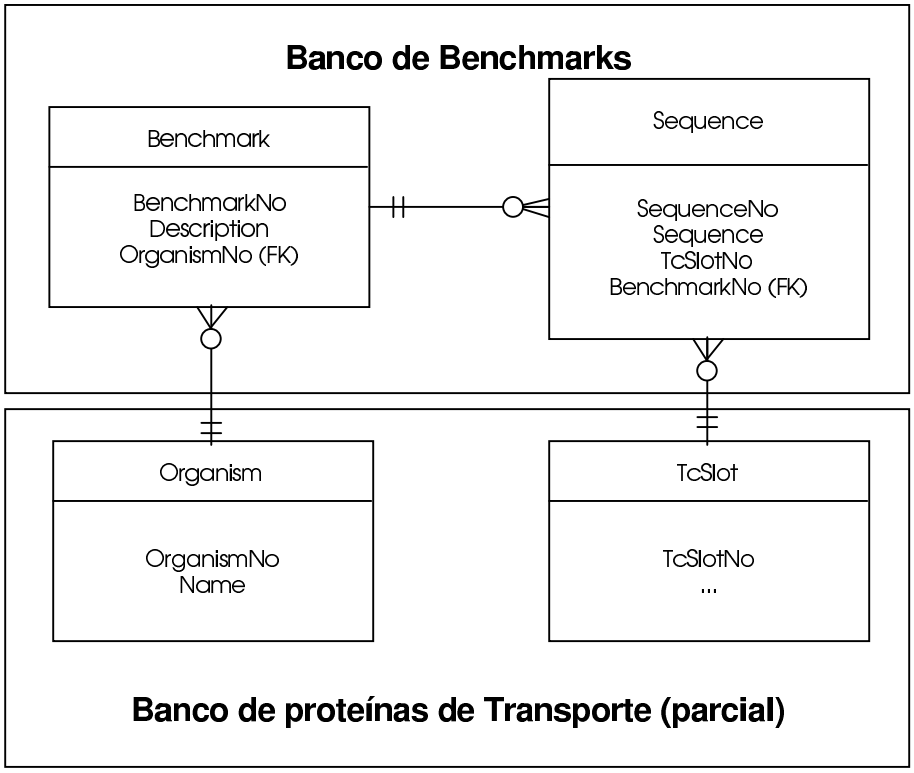

Figura 6: DER do banco de benchmarks e sua relação com o banco de transporte (Figura 4).

A Layout da Tabelas

A.1 Tabelas do Sistema Atual

A.1.1 GuiaBlast

Campo	**Tipo**	**Descrição**
GuiaNo	int(10)	Número interno de identificação do registro na tabela Guia.
Example	varchar(50)	Nome da seqüência exemplo.
WorstScoreIn	double(16,4)	O pior score dentre as seqüências do banco do BLAST que bateram com o exemplo (*hit*).
WorstEValueIn	double(16,4)	Versão evalue do *WorstScoreIn*.
WorstInFamilyScore	double(16,4)	Pior score do *hit* dentro da mesma família que o primeiro *hit*. Considera-se, neste caso, um bloco de hits consecutivos que pertencem à mesma família.
WorstInFamilyEValue	double(16,4)	Versão evalue do *WorstInFamilyScore*.
BestOutFamilyScore	double(16,4)	Melhor score do hit **fora** do primeiro bloco (bloco de hits com mesma família, em o primeiro hit é o melhor de todos os resultados).
BestOutFamilyEValue	double(16,4)	Versão evalue do *BestOutFamilyScore*.
BestHitScore	double(16,4)	Melhor score de todos os resultados (primeiro hit).
BestHitEValue	double(16,4)	Versão evalue do *BestHitScore*.
NTopsInFamily	int	Número de elementos no primeiro bloco.
Sequence	text	Seqüência de aminoácidos da proteína deste exemplo.
Family	varchar(20)	Família ao qual pertence este exemplo.

Tabela 4: Definição das tabelas do tipo Guia para resultados do Blast puro.

A.1.2 GuiaPsi

Campo	Tipo	Descrição
GuiaNo	int(10)	Número interno de identificação do registro na tabela.
Example	varchar(50)	Nome da seqüência exemplo.
Converged	char(1)	"Y" se o Psi-Blast convergiu no final, caso contrário "N".
Rounds	int	Quantas iterações aconteceram até o Psi-Blast parar.
WorstScoreIn	double(16,4)	O pior *score* dentre as seqüências do banco do BLAST que bateram com o exemplo (*hit*) e que ficaram dentro da faixa de consolidação.
WorstEValueIn	double(16,4)	Versão *evalue* do *WorstScoreIn*.
BestScoreOut	double(16,4)	O melhor *score* dentre as seqüências do banco do BLAST que bateram com o exemplo (*hit*) e que ficaram fora da faixa de consolidação.
BestEValueOut	double(16,4)	Versão *evalue* do *BestScoreOut*.
WorstInFamilyScore	double(16,4)	Pior score do hit dentro da mesma família que o primeiro hit. Considera-se, neste caso, um bloco de hits consecutivos que pertencem à mesma família.
WorstInFamilyEValue	double(16,4)	Versão *evalue* do *WorstInFamilyScore*.
BestOutFamilyScore	double(16,4)	Melhor score do hit **fora** do primeiro bloco (bloco de hits com mesma família, em o primeiro hit é o melhor de todos os resultado).
BestOutFamilyEValue	double(16,4)	Versão *evalue* do *BestOutFamilyScore*.
BestHitScore	double(16,4)	Melhor score de todos os resultados.
BestHitEValue	double(16,4)	Versão *evalue* do *BestHitScore*
NTopsInFamily	int	Número de elementos no primeiro bloco.
Sequence	text	Seqüência de aminoácidos da proteína deste exemplo.
Family	varchar(20)	Família ao qual pertence este exemplo.

Tabela 5: Definição das tabelas do tipo Guia para resultados do Psi-Blast. Ilustração de boa parte dos dados pode ser encontrada na Figura 2.

A.1.3 ThresholdPsi ou ThresholdBlast

Campo	Tipo	Descrição
ThresholdNo	int	Número interno de identificação do registro na tabela do tipo Threshold.
Family	varchar(20)	O número TC da família que deve usar estes *Threshold*s.
NTop	int	Ver definição do parâmetro N na seção 2.3.1, página 6.
Score	double(16,4)	Ver definição do parâmetro S na seção 2.3.1, página 6.

Tabela 6: Definição das tabelas do tipo Threshold

A.1.4 GabaritoXylella

Campo	Tipo	Descrição
Example	varchar(50)	Nome da seqüência de proteína
Family	varchar(20)	número TC da família

Tabela 7: Definição das tabelas do tipo Gabarito

A.2 Tabelas do Sistema Novo

A.2.1 Category

Campo	Tipo	Descrição
CategoryNo	int(10)	Número seqüencial interno de identificação da tabela.
CategoryTc	char(15)	Código TC da categoria. Ex.: *1.A, 1, 1.A.23.*
Name	char(100)	Nome da categoria. Exemplo: *a-type channels*, *The Major Intrinsic Protein (MIP) Family.*
Level	int	Nível da categoria: 1, 2, 3 ou 4. O último nível pode ser 3 ou 4.
Acronym	char(20)	Normalmente em nível de família (último nível), este dado é um nome curto que vem entre parênteses do nome de forma detalhada. Ex.: *TRP-CC, MscS, NaT-MMM.*
UpperNo	int(10)	CategoryNo "pai" desta categoria.

Tabela 8: Definição da tabela **Category**

A.2.2 TcSlot

Campo	Tipo	Descrição
TcSlotNo	int(10)	Número seqüencial interno de identificação da tabela.
CategoryNo	int(10)	CategoryNo que representa a família deste TC. Serão apenas categorias de nível 3 ou 4.
TcComplement	char(15)	O complemento do código TC, que junto com o CategoryTc da família formará o número TC completo. O número TC é único. Por exemplo, se neste campo tivermos 1.1 e o CategoryTc da família era 3.*A*.9, então o número TC completo é 3.*A*.9.1.1.
Name	char(100)	Nome deste TC *slot* . Refere-se a um tipo de proteína ou sistema. Exemplo: *The competence-related pilin exporter, ComGA/GB*.

Tabela 9: Definição da tabela **TcSlot**.

A.2.3 Source

Campo	Tipo	Descrição
SourceNo	int(10)	Número seqüencial interno de identificação da tabela.
Name	char(100)	Origem (reino ou classe) onde são encontradas as proteínas classificadas em determinado número TC. Alguns exemplos de origem: animals, bacteria, eukaryotic protista, Sinorhizobium meliloti, plants, yeast.

Tabela 10: Definição da tabela **Source**

A.2.4 TcSlotSource

Campo	Tipo	Descrição
TcSlotNo	int(10)	O código interno do número TC.
SourceNo	int(10)	O código interno na tabela Source correspondente ao local onde as proteínas relacionadas ao TC são encontradas.
Comments	text	Texto adicional relacionado à ligação *TcSlot* e *Source*.

Tabela 11: Definição da tabela **TcSlotSource**. Relacionamento entre as tabelas *TcSlot* e *Source*.

A.2.5 Example

Campo	Tipo	Descrição
ExampleNo	int(10)	Número seqüencial interno de identificação da tabela.
Name	char(100)	Nome geral do exemplo dado. Pode conter várias proteínas agregadas ao exemplo, neste caso trata-se de um sistema de proteínas. Há vários ítens na família *3.A.1.1 — The ATP-binding Cassette (ABC) Superfamily* — ilustrando este caso.

Tabela 12: Definição da tabela **Example**.

A.2.6 TcSlotExample

Campo	Tipo	Descrição
TcSlotNo	int(10)	O código interno do número TC.
ExampleNo	int(10)	O código interno do exemplo na tabela *Example* que pertence ao TcSlot.

Tabela 13: Definição da tabela **TcSlotExample**. Relacionamento entre as tabelas *TcSlot* e *Example*.

A.2.7 Protein

Campo	Tipo	Descrição
ProteinNo	int(10)	Número seqüencial interno de identificação da tabela.
Name	char(100)	Nome da proteína.
OrganismNo	int(10)	O organismo que possui a seqüência desta proteína exemplo em questão.

Tabela 14: Definição da tabela **Protein**.

A.2.8 Organism

Campo	Tipo	Descrição
OrganismNo	int(10)	Número seqüencial interno de identificação da tabela.
Name	char(100)	Nome do organismo. Um organismo pode ser determinadas plantas, animais, bactérias e assim por diante.

Tabela 15: Definição da tabela **Organism**.

A.2.9 Role

Campo	Tipo	Descrição
RoleNo	int(10)	Número seqüencial interno de identificação da tabela.
Name	char(100)	Papel desempenhado por uma proteína em um sistema.

Tabela 16: Definição da tabela **Role**.

A.2.10 ExampleProtein

Campo	Tipo	Descrição
ExampleNo	int(10)	O exemplo/sistema especificado.
RoleNo	int(10)	Código interno que identifica o papel que a proteína desempenha no sistema identificado em ExampleNo, se for o caso.
ProteinNo	int(10)	Proteína que faz parte do exemplo/sistema dado.

Tabela 17: Definição da tabela **ExampleProtein**. Ligação entre as tabelas *Example*, *Protein* e *Role*.

A.2.11 Sequence

Campo	**Tipo**	**Descrição**
SequenceNo	int(10)	Número seqüencial interno de identificação da tabela.
Sequence	text	A seqüência de aminoácidos da proteína .
TMS	int	Quantidade de regiões transmembranas.
AccessionSP	char(10)	O código de acesso a essa seqüência no banco do SWISS-PROT.
AccessionGI	char(10)	O código de acesso a essa seqüência no banco do GenBank.
AccessionGB	char(10)	O código de acesso a uma entrada do GenBank, que contém esta seqüência, mas também pode conter outras seqüências adicionais.
GeneName	char(50)	Nome do gene
Update	date	data de última atualização da seqüência
ProteinNo	int(10)	Identificação da proteína na tabela *Protein* que tem essa seqüência

Tabela 18: Definição da tabela **Sequence**.

Referências

[1] Stephen F. Altschul, Thomas L. Madden, Alejandro A. Schäffer, et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search. *Nucleic Acids Research*, 25(17):3389–3402, 1997.

[2] João Meidanis, Marília D. V. Braga, and Sérgio Verjovski-Almeida. Whole genome analysis of transporters in the plant pathogen *Xylella fastidiosa*. *Microbiology and Molecular Biology Review*, 2002. To appear.

[3] Mysql. Web site, January 2002. http://www.mysql.com/.

[4] Milton Saier. Transport protein overview. Web site, September 2001. http://www-biology.ucsd.edu/ msaier/transport/.