Universidade Estadual de Londrina

MARCIO RODRIGO SANTOS

VNBLAST - SISTEMA DE GERENCIAMENTO DO NETBLAST

Londrina
2011

MARCIO RODRIGO SANTOS

VNBLAST - SISTEMA DE GERENCIAMENTO DO NETBLAST

Trabalho de Dissertação de Mestrado apresentado ao Departamento de Computação, UEL - Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Wesley Attrot.

Londrina
2011

Catalogação Elaborada pela Divisão de Processos Técnicos da Biblioteca Central da Universidade Estadual de Londrina

Dados Internacionais de Catalogação-na-Publicação (CIP)

S237v Santos, Marcio Rodrigo.
VNBlast : sistema de gerenciamento do Netblast / Marcio Rodrigo Santos. – Londrina, 2011. 108 f.: il.

Orientador: Wesley Attrot.
Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2011.
Inclui bibliografia.

1. Software – Desenvolvimento – Teses. 2. Ferramentas de busca na Web – Teses. 3. Bioinformática – Teses. 4. Serviços da Web – Teses. 5. Gerenciamento de configurações de software – Teses. I. Attrot, Wesley. II. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDU 519.68.02

MARCIO RODRIGO SANTOS

VNBLAST - SISTEMA DE GERENCIAMENTO DO NETBLAST

Trabalho de Dissertação de Mestrado apresentado ao Departamento de Computação, UEL - Universidade Estadual de Londrina, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

_____
Orientador. Prof. Dr. Wesley Attrot
Universidade Estadual de Londrina - UEL

_____
Prof. Dr. Emerson José Venancio
Universidade Estadual de Londrina - UEL

_____
Prof. Dr. Alan Salvany Felinto
Universidade Estadual de Londrina - UEL

_____
Prof. Dr. Jacques Duílio Brancher
Universidade Estadual de Londrina - UEL

Londrina, 18 de Novembro de 2011.

Agradecimentos

Agradeço a Deus por ter sido o criador, redentor, mantenedor, guia e acima de tudo um pai amoroso e fiel.

Agradeço em especial ao meu orientador, Prof. Dr. Wesley Attrot, pelo apoio, amizade, direção e paciência em todos os momentos.

Agradeço à minha esposa Gisele e meus filhos Johan e Arthur (in memorian), por terem sido compreensivos durante esta árdua caminhada.

Agradeço aos meus pais Genciano e Sueli e minha irmã Karla, pelas constantes e fervorosas preces que foram decisivas para as vitórias alcançadas.

Agradeço aos colegas e amigos da UEL, em especial ao Prof. Dr. Jacques Duílio Brancher, ao Prof. Dr. Alan Salvany Felinto e ao Prof. Dr. Rodolfo Miranda de Barros, por acreditarem e apoiarem meu trabalho.

Agradeço à ADOBE por ter gentilmente cedido uma licença de uso da IDE de desenvolvimento do Flex para o desenvolvimento do VNBlast..

SANTOS, Marcio. **VNBlast - Sistema de Gerenciamento do Netblast**. 2011. 108 f. Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Londrina. 2011.

RESUMO

O GenBank é um banco de dados público de sequências de nucleotídeos, atualmente gerido pelo NCBI - National Center for Biotechnology Information, que fornece mecanismos para o acesso e processamento de informações armazenadas. Uma forma de acesso às informações do GenBank é através da suíte BLAST (Basic Local Alignment Search Tool) para busca de similaridade local entre sequências genéticas. As informações do GenBank podem ser acessadas pelo website do NCBI ou localmente. O website NCBI BLAST é uma maneira fácil de encontrar sequências, mas impõe algumas limitações na parametrização da consulta, e pesquisas em lote de larga escala não estão disponíveis. Objetivando preencher esta lacuna, este trabalho irá apresentar a ferramenta VNBlast. O VNBlast é uma aplicação web amigável que tem como base o NetBlast do NCBI, que oferece um número substancial de parâmetros para a busca e alinhamento de sequências que são efetuadas diretamente no GenBank através de webservices, evitando a necessidade de download dos conjuntos de dados e apresentando como resultado informações constantemente atualizadas. O VNBlast foi capaz de executar alinhamentos de forma mais simplificada e intuitiva que o Netblast, apresentou maiores possibilidades de resultados que o site BLAST da NCBI, ofereceu mais parâmetros para busca direta que as ferramentas citadas e possibilitou a execução de buscas em lote.

Palavras-Chaves: VNBlast. BLAST. GUI. Bioinformática.

SANTOS, Marcio. **VNBlast - A Netblast Management System**. 2011. 108 p. Dissertation (Master's degree) - State University of Londrina, Londrina. 2011.

ABSTRACT

GenBank is a public database of nucleotide sequences built by the National Center for Biotechnology Information (NCBI) that provide mechanisms for accessing and processing stored information. A way to access information on GenBank is BLAST (Basic Local Alignment Search Tool) a program to search for local similarity between sequences. GenBank information can be accessed through NCBI website or locally. NCBI BLAST website is an easy way to find sequences, but imposes some limitations in the parameterization of the query, and batch searches in large-scale are not available. In order to fill this gap, this paper will introduce the VNBlast tool. VNBlast is a user-friendly web application based on NCBI NetBlast, which offers a substantial number of parameters for search and alignment of sequences directly on GenBank, avoiding the need of manual download of datasets and always presenting as result updated information. VNBlast was able to perform alignments in an easier and more intuitive way than Netblast, it presented more possibilities of results than the NCBI BLAST website, offered more parameters for direct search that the mentioned tools and allowed the execution of batch searches.

Keywords: VNBlast. BLAST. GUI. Bioinformatics.

Lista de Figuras

Lista de Siglas e Abreviaturas

NCBI	National Center for Biotechnology Information
BLAST	Basic Local Alignment Search Tool
GUI	Graphics User Interface
EMBL	European Molecular Biology Laboratory
DDBJ	DNA DataBank of Japan
PDB	Protein Data Bank
ENTREZ	Global Query Cross-Database Search System
VNBlast	Visual Netblast
PBS	Portable Batch System
HSP	High-Scoring Pairs
XML	Extensible Markup Language
GO	Gene Ontology
KEGG	Kyoto Encyclopedia of Genes and Genomes
SMART	Simple Modular Architecture Research Tool
SQL	Structured Query Language
RDBMS SQL	Code by the Relational Database Management System
ID	Identif er
GI	Unique ID for Genbank Records
Mrna	Messenger RNA
rRNA	Ribossomal RNA
EBI	European Bioinformatics Institute
RDP	Ribosomal Database Project
MSP	Maximal Segmented Pair
EJB	Enterprise Java Beans
API	Application programming Interface
Java SE	Java Standard Edition
EE	Enterprise Edition
JCP	Java Community Process
J2EE	Java2 Enterprise Edition
HP	Hewlett-Packard
JDO	Java Data Objects

AS	Application Server
JMS	Java Message Service
JMX	Java Management Extensions
R.I.A.	Rich Internet Applications
ECMA	European Computer Manufacters Association
O.O.	Object Oriented
SDK	Software Development Kit
MXML	Macromedia eXtensible Markup Language
HTML	HyperText Markup Language
CDD	Conserved Domains Database
PRF	Protein Research Foundation
EST	Expressed Sequence Tag

Sumário

1 INTRODUÇÃO

As pesquisas e aplicações práticas da biologia molecular têm sido muito importantes para uma melhor compreensão da vida e das doenças (SOH et al., 2007). A compreensão do funcionamento celular em nível molecular possibilita um melhor entendimento do organismo como um todo (AMARAL; SANTELLI, 2011). Visando aprimorar o conhecimento biomolecular, pesquisas em larga escala têm sido empregadas na descrição do genoma de uma ampla variedade de organismos, destacando-se entre elas o próprio projeto do genoma humano (RIDLEY, 2006; NG; PANG, 2010).

O sequenciamento de genomas e estratégias de sequenciamento genético têm gerado grandes volumes de sequências genéticas anotadas, tanto de nucleotídeos quanto de proteínas. A melhor maneira de tornar estas sequências de nucleotídeos, proteínas e outras informações genômicas disponíveis para pesquisadores, estudantes e demais interessados em todo o mundo, tem sido através de bases de dados públicas que permitem o acesso facilitado a estas informações.

Dentre os vários bancos de dados de informações biomoleculares atualmente disponíveis, o GenBank (BENSON et al., 2009), que é um banco de dados público de sequências de nucleotídeos e proteínas, contém informações genéticas de mais de 380 mil organismos e é mantida pelo NCBI - *National Center for Biotechnology Information* (BENSON et al., 2009) desde a década de 80, tendo como sede a cidade de Bethesda, no estado de Maryland - Estados Unidos. Este banco foi criado a partir de sequências nucleotídicas obtidas por pesquisadores oriundos de vários lugares do mundo (BENSON et al., 2009).

As informações atualmente contidas no GenBank são provenientes não apenas da submissão direta de novas sequências genéticas obtidas através do processo de sequenciamento por pesquisadores e laboratórios, mas também através de sequências submetidas por outros bancos de dados interconectados ao GenBank e distribuídos pelo mundo.

Dentre eles, é possível citar o europeu EMBL - *European Molecular Biology Laboratory* (STOESSER et al., 2002), um banco de dados descentralizado, distribuído entre

algumas localidades da Europa; o japonês DDBJ - *DNA Data Bank of Japan* (TATENO et al., 2000) e o intercontinental PDB - *Protein Data Bank* (BERMAN et al., 2000). Todos estes bancos estão interligados e compartilham informações entre si diariamente.

Nestes bancos de dados genéticos, existem aproximadamente três décadas de informações armazenadas e segundo (BENSON et al., 2009), no fim de 2009 o número de bases de nucleotídeos era de aproximadamente 106 bilhões, totalizando aproximadamente 108 milhões de sequências individuais. Devido a este crescimento exponencial, estima-se que o volume de informações do GenBank dobra a cada 35 meses.

Mas tão importante quanto o armazenamento adequado destas informações em bases de dados, assim é o processo de localização de informações específicas em meio a este grande volume de dados. O processo de comparação entre informações genéticas e dados biomoleculares previamente armazenados nestes bancos de dados genéticos possibilitam uma melhor compreensão das características determinantes da nova sequência obtida, possibilitando ao pesquisador definir com maior precisão a descrição que acompanhará a nova sequência. Esta comparação entre sequências é feita pelo processo conhecido como alinhamento de sequências genéticas.

Estes grandes bancos de dados, tais como o GenBank, disponibilizam de forma pública e livre de ônus mecanismos para o acesso, localização e processamento de dados armazenados em seus arquivos. No GenBank, as informações podem ser acessadas pelo endereço de internet do NCBI ou localmente, através do *download* manual dos arquivos de dados disponíveis por ftp, em conjunto com aplicativos em modo texto para o processamento e pesquisa das informações contidas nos arquivos de dados.

Existem atualmente dois mecanismos principais de acesso à informações no GenBank, disponibilizados pelo próprio NCBI: o ENTREZ (SCHULER et al., 1996) e o BLAST (ALTSCHUL et al., 1997). O ENTREZ é um sistema de acesso ao GenBank que suporta a busca de identificadores de texto e anotações de sequências, que possibilita a obtenção de informações em diferentes formatos. O ENTREZ baseia-se no fato de que existem relações lógicas pré-existentes entre as entradas individuais encontradas em numerosas bases de dados públicas.

A existência de tais conexões naturais, principalmente de natureza biológica, fomentou o desenvolvimento de um método através do qual todas as informações sobre uma determinada entidade biológica poderiam ser encontradas, sem a necessidade de consulta e análise sequencial em bancos de dados distintos. Estas necessidades corroboraram para o surgimento da ferramenta ENTREZ, um metabuscador desenvolvido e mantido pelo NCBI.

Já o BLAST - *Basic Local Alignment Search Tool* é um algoritmo de alinhamento genético, implementado como uma aplicação que busca similaridade local entre sequências genéticas e têm sido amplamente utilizado para fazer inferências sobre a função de uma determinada sequência e suas possíveis relações filogenéticas. O BLAST produz como resultado o alinhamento local de sequências, ou seja, apenas uma parte de cada sequência deverá ser alinhada. O BLAST faz uso da teoria estatística para determinar se um acerto pode ter ocorrido por acaso ou não.

Este padrão de algoritmo heurístico no qual o BLAST foi desenvolvido é conhecido como método de alinhamento local (EDDY, 2008). Algoritmos heurísticos de alinhamento de sequências têm como uma de suas principais características a velocidade no processo de alinhamento. Estes algoritmos são recomendados para alinhamento entre sequências de tamanhos diferentes ou sequências com apenas alguns trechos conservados.

O algoritmo BLAST se subdivide em três etapas básicas. A primeira etapa cria uma lista de sequências curtas (palavras) com pontuação acima do valor limite do programa, quando alinhadas com a sequência alvo. Na segunda etapa, o banco de dados é consultado para serem obtidas as ocorrências destas palavras no mesmo. Por fim as palavras são combinadas primeiramente em pares, estendidos a alinhamentos locais entre o alvo e cada sequência do banco de dados, que serão classificados conforme uma pontuação estabelecida de acordo com o número de palavras alinhadas.

Uma importante aplicação do BLAST é a identificação de regiões conservadas em sequências. Esta identificação têm por objetivo a projeção de sondas específicas que serão utilizadas para as reações de detecção molecular ((MARSHALL, 2004); (WECKX et al., 2004); (SCHRETTER; MILINKOVITCH, 2005); (BALLY et al., 2010).

No entanto, devido ao crescimento exponencial do número de sequências disponíveis no GenBank, a busca de informações e o processo de alinhamento muitas vezes demandam um tempo proibitivo. Além disso, a página de internet do programa BLAST, acessível através do endereço de internet do NCBI, apresenta um número reduzido de parâmetros disponíveis para consultas e alinhamentos. Em adição a estas limitações, buscas e alinhamentos de sequências em lote não estão disponíveis.

Em contrapartida, o NCBI disponibiliza um conjunto de ferramentas para buscas, alinhamentos locais e remotos, desenvolvidos e distribuídos como aplicações em modo texto, que possuem grande capacidade e desempenho de processamento. Dentre estas aplicações destaca-se a família de programas BLAST (Blastn, Blastp, Blastx, Tblastn, Tblastx, Mega BLAST e BLAST Psi) (ALTSCHUL et al., 1997).

Boa parte destas ferramentas foi desenvolvida para a realização de pesquisas e alinhamentos de sequências obtidas no GenBank através de bases de dados locais, com exceção de uma aplicação cliente chamada Netblast. O Netblast equivale ao conjunto de ferramentas mencionadas anteriormente, mas efetua as buscas e alinhamentos diretamente no GenBank, sem a necessidade de prévio *download* dos arquivos de dados.

Apesar da capacidade e do desempenho que as aplicações console disponibilizadas pelo NCBI possuem, às vezes, a linha de comando necessária para a realização de uma consulta pode se tornar extensa, dependendo do número de parâmetros necessários para sua execução, aumentando as chances de erros de digitação e a consequente falha no processo de busca e alinhamento. Esta falha poderá ocorrer na forma de uma mensagem de erro ou ainda através de um resultado de busca indesejado.

Objetivando possibilitar uma melhor interatividade com o usuário do que as versões BLAST em modo texto; oferecer um maior número de parâmetros de busca e alinhamento e adicionalmente a opção de buscas em lote quando em comparação com a versão BLAST para a internet da NCBI; e, de modo geral, uma maior facilidade no processo de busca e alinhamento, este trabalho irá apresentar o VNBlast, uma versão gráfica para a internet do Netblast, aplicativo este que foi originalmente desenvolvido para utilização por linhas de comando, mas que agora será apenas o núcleo de uma interativa interface gráfica amigável para a internet.

1.1 A dissertação

Para este fim, o presente trabalho será segmentado da seguinte forma: o Capítulo 2 apresenta os trabalhos correlatos, o Capítulo 3 apresenta o alinhamento de sequências através do algoritmo BLAST, o Capítulo 4 apresenta a ferramenta VNBlast, o Capítulo 5 apresenta a proposta de trabalhos futuros e a conclusão.

2 Trabalhos Correlatos

A biologia possui uma vasta gama de possibilidades de aplicação de técnicas de computação para a resolução de problemas tipicamente biológicos em um tempo mais adequado do que por métodos manuais. Um exemplo da aplicação de técnicas computacionais na área biológica é o procedimento de comparação entre sequências genéticas, conhecido como alinhamento de sequências genéticas. Este procedimento, quando efetuado manualmente, torna-se lento e altamente suscetível a erros por falha humana.

Com o surgimento da bioinformática, vários algoritmos computacionais foram desenvolvidos visando o aprimoramento da técnica de alinhamento de sequências. Com o passar dos anos vários destes algoritmos sofreram atualizações visando um melhor desempenho e correções de possíveis falhas. Da mesma forma, vários programas abertos surgiram como implementações dos algoritmos desenvolvidos, facilitando o acesso a estas ferramentas para pesquisadores e estudantes. Este capítulo apresentará alguns programas dedicados a este fim.

2.1 BLAST via Website do NCBI

O NCBI disponibiliza uma forma de acesso simplificada ao mecanismo de busca e alinhamento BLAST, através da própria página do NCBI. Segundo a definição dada pelo próprio NCBI, o objetivo do BLAST é encontrar regiões de similaridade local entre sequências. Este programa compara sequências de nucleótidos ou proteínas com as bases de dados de sequências e calcula a significância estatística dos alinhamentos. O BLAST pode também ser usado para inferir relações funcionais e evolutivas entre as sequências, assim como ajudar a identificar os membros de famílias de genes. A Figura 2.1 apresenta a página do NCBI contendo o mecanismo básico de consultas ao GenBank. Já a Figura 2.2 apresenta a opção avançada de seleção de parâmetros para a consulta ao GenBank. Apesar da facilidade de utilização do site disponibilizado pelo NCBI, este apresenta algumas limitações em sua utilização. Entre elas estão o número limitado de parâmetros de busca e alinhamento e a impossibilidade da execução de buscas em lote de larga escala. Isto se deve ao fato de que o NCBI recebe uma

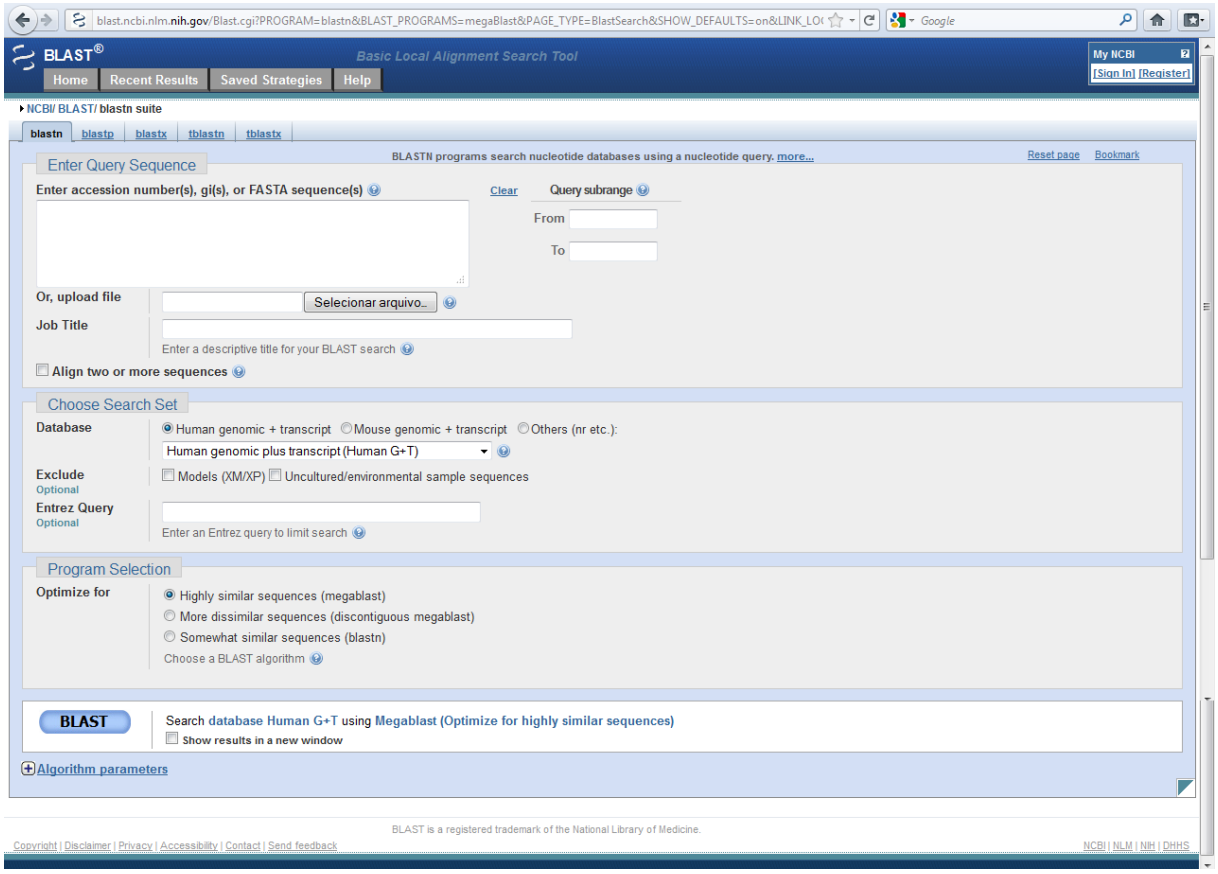


Figura 2.1: Website da NCBI para buscas básicas usando o BLAST.

grande quantidade de acessos simultâneos diários, oriundos de várias partes do mundo. Estas restrições possibilitam uma maior estabilidade na disponibilização deste serviço à comunidade científica.

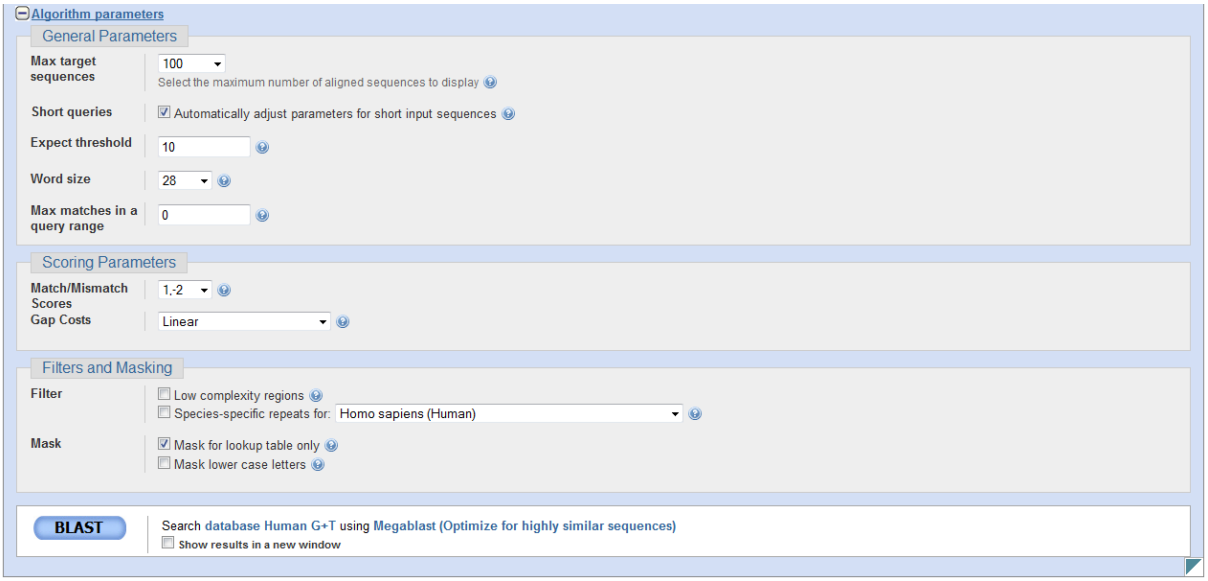


Figura 2.2: Website da NCBI exibindo a opção de parâmetros avançados de busca usando o BLAST.

2.2 Alkahest NuclearBLAST

O Alkahest NuclearBLAST (DIENER et al., 2005) é uma interface gráfica de internet para gerenciamento e análise de sequências, que foi desenvolvida como um aplicativo servidor. O pacote de instalação deste software fornece uma base de dados MySQL para armazenar as consultas realizadas. Através das informações do banco de dados, obtidas a partir de consultas usando a ferramenta Blastall (ALTSCHUL et al., 1997) é possível fazer a análise e mineração de resultados.

O principal objetivo para o desenvolvimento do NuclearBLAST foi o de fornecer aos biólogos um sistema centralizado, onde os resultados gerados pelo algoritmo BLAST pudessem ser obtidos e recuperados através de um sistema de armazenamento de dados relacional para análise comparativa. Um outro objetivo secundário no desenvolvimento desta ferramenta foi o de projetar uma solução que fosse totalmente desenvolvida por ferramentas de código aberto.

Além dos recursos do banco local, o NuclearBLAST pode fornecer os termos Ontológicos (*Ontology*) do gene para sequências e opções para exportação de informações em formato de planilha. A fonte de informação utilizada pelo NuclearBlast advém dos conjuntos de dados obtidos através do pacote BLAST, porém sua utilização é feita através de uma base de dados local, que é alimentada através de arquivos baixados manualmente do endereço de ftp do NCBI.

Estes conjuntos de dados baixados irão alimentar o banco de dados local para pesquisas. O uso de um banco de dados local resulta em um bom desempenho para buscas sucessivas, mas também pode gerar informações desatualizadas, se o banco de dados não for constantemente atualizado.

O NuclearBLAST tira proveito dos recursos da computação em cluster, para aumentar a taxa de transferência de grandes volumes de dados do BLAST. O desenvolvimento dessa ferramenta foi baseado na utilização de uma série de pacotes de software aberto, incluindo o BioPerl (COMMUNITY, 2011), o Apache (FOUNDATION, 2011), o PHP (TEAM, 2011e) e o MySQL (TEAM, 2011b), assim como o uso do Linux como sistema operacional base.

Os navegadores da internet foram escolhidos como interface para o NuclearBLAST, pois são amplamente utilizados e globalmente disponíveis para várias plataformas de sistemas operacionais. O servidor de internet Apache foi utilizado para fornecer controle de acesso e conexões criptografadas, caso o usuário necessite assegurar a confidência na sua pesquisa.

No NuclearBLAST, o BioPerl fornece um módulo de análise para os resultados do BLAST, que são carregados posteriormente para o banco de dados MySQL. Todas as informações sobre as pesquisas do BLAST são carregadas no banco de dados e o status de cada consulta em um trabalho é monitorado pela aplicação.

Este projeto cliente-servidor do programa permite o uso de vários computadores executando as análises BLAST em um cluster, empregando para isso o software de gerenciamento PBS (Sistema de Lotes Portável) (TEAM, 2011d).

O NuclearBLAST pode ser utilizado tanto em um único computador quanto em vários computadores em um laboratório, utilizados como nós de trabalho e sendo executados em seu tempo livre de processamento ou em uma rede de computação dedicada a esta tarefa. A instalação mínima do NuclearBLAST requer uma estação de trabalho típica que atuará tanto como servidor quanto como estação de trabalho.

A fim de manter o banco de dados MySQL com um tamanho gerenciável e reduzir dados redundantes no sistema, foi adotado por padrão o formato de banco de dados BLAST como o único formato de entrada aceito. Apenas os nomes dentro de cada conjunto de dados e uma quantidade mínima de metadados são armazenados no banco de dados MySQL. Outra forma de minimizar o tamanho do banco de dados utilizado, foi através do armazenamento de todas as estatísticas de resultados no banco de dados, porém excluindo-se os alinhamentos reais. Quando necessário, os alinhamentos são recriados, extraindo-se as duas

sequências do banco de dados local do BLAST e alinhando-as através do Bl2seq (ALTSCHUL et al., 1997). Os usuários podem importar conjuntos de dados, análises, e ver os resultados na interface de internet PHP através de seu navegador. Parâmetros de linha de comando podem ser utilizados para o desenvolvimento de consultas mais complexas, ou para fins especiais.

Os arquivos de sequência utilizados são importados em formato FASTA e o NUCLEAR-BLAST faz uso do utilitário Formatdb do NCBI para transformar sequências no formato exigido pelo Blastall. Ao importar um conjunto de dados de sequências no Nuclear-BLAST, o usuário especifica se a sequência pode ser usada em pesquisas subsequentes como uma consulta, um alvo, ou ambos. Especificando a opção ambos, o programa permite análises BLAST recíprocas entre conjuntos de dados menores.

Estes parâmetros podem ser ajustados para permitir aos usuários o acesso a grandes conjuntos de dados como alvos, mas não como consultas. Isso evita que erros ocasionais de parâmetros impliquem na consulta equivocada de bases de dados muito grandes, como o banco de dados GenBank de proteínas não redundantes, ao invés de conjuntos de dados muito menores (e não vice-versa).

Um assistente de pedidos de trabalho orienta o usuário durante o processo de montagem de uma consulta. A Figura 2.3 apresenta o fluxograma de tarefas no NuclearBLAST. Esta ilustração de fluxo de trabalho do programa descreve o progresso de uma solicitação de análise BLAST.

A Figura 2.4 apresenta imagens do Assistente para Solicitação de tarefas no NuclearBLAST. O primeiro painel mostra o conjunto de definições de consultas possíveis no sistema. Dependendo da escolha, os submenus adequados são disponibilizados, conforme pode ser visto no segundo quadro da Figura 2.4. Na sequência o assistente permite limitar os valores de resultados armazenados e dá opções para a seleção do banco de dados BLAST de destino no sistema.

A Figura 2.5 apresenta cópias de tela da interface do NuclearBLAST. O primeiro painel da Figura 2.5 mostra os trabalhos solicitados e seu progresso. Um clique do mouse em um *hyperlink* de um respectivo trabalho resulta na exibição de uma página, apresentada no segundo painel da Figura 2.5, contendo os resultados da consulta de sequências múltiplas dentro do trabalho solicitado. O *link* nesta página que mostra o número de acessos trará a localização e as estatísticas de acessos múltiplos para a sequência de consulta, conforme apresenta o terceiro painel da Figura 2.5. Ao se clicar em um acerto, o sistema fornece uma visão dos nomes selecionados e exibe estatísticas para este acerto único, conforme apresenta o quarto painel da Figura 2.5. O hyperlink das sequências apresenta uma página com informações

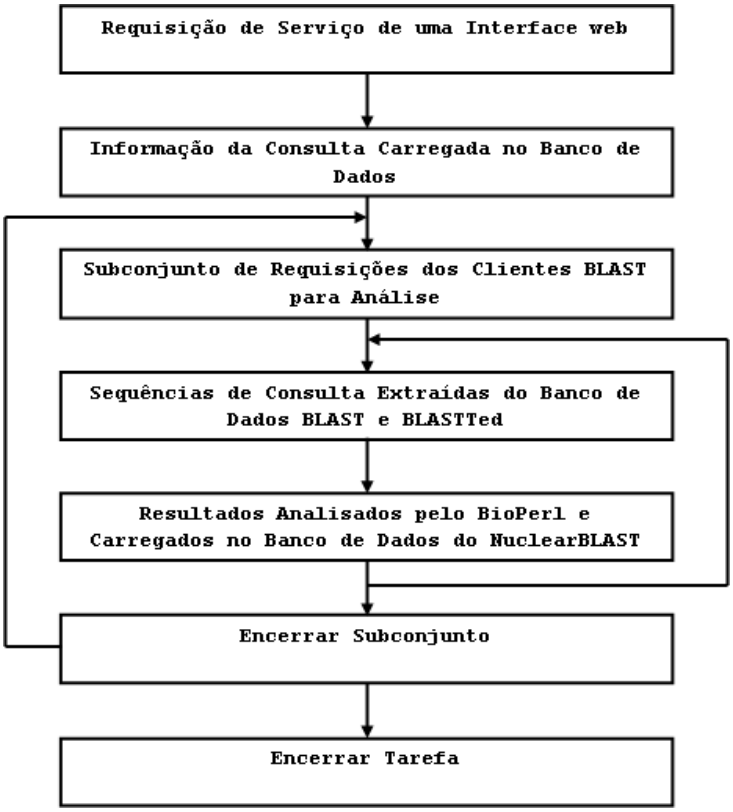


Figura 2.3: Fluxograma de tarefas no NuclearBlast.

sobre a sequência particular.

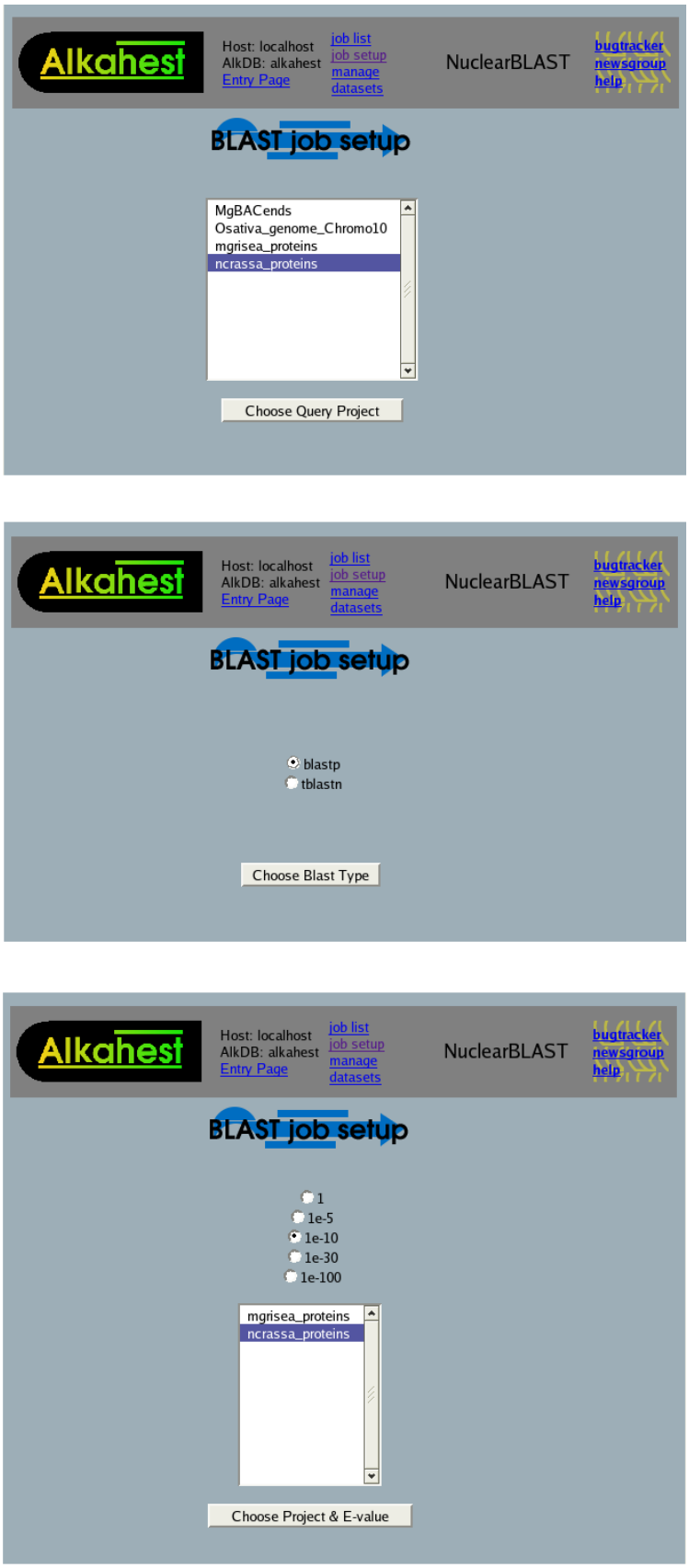


Figura 2.4: Cópias de tela do assistente de consultas do NuclearBlast.

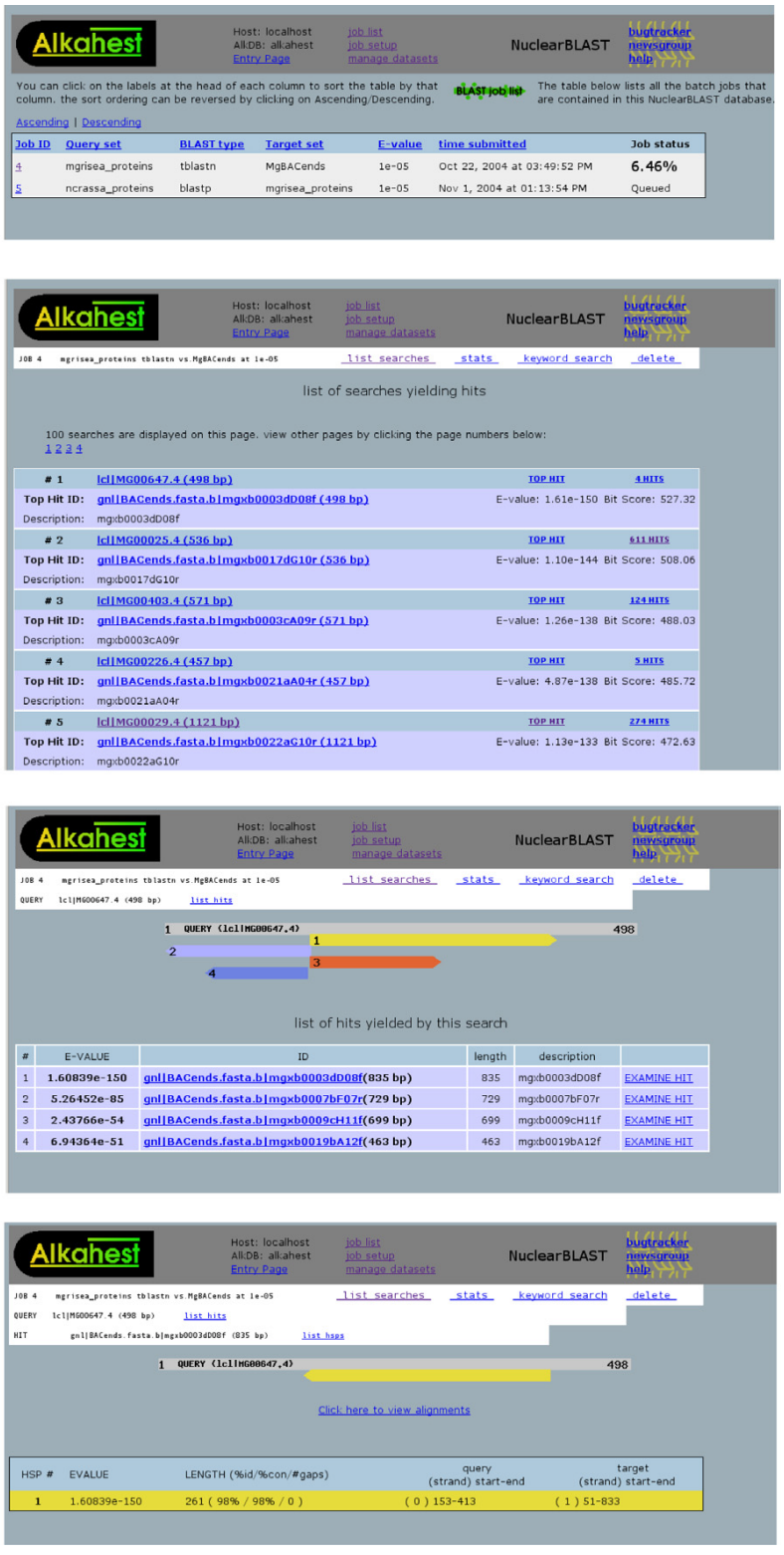


Figura 2.5: Cópias de tela da interface para internet do NuclearBlast.

2.3 NOBLAST e JAMBLAST

O NOBLAST (LAGNEL; TSIGENOPOULOS; LLIOPOULOS, 2009) é um programa de código aberto que fornece um formato de saída tabular para vários programas BLAST NCBI sem a necessidade de uso de nenhum analisador intermediário e fornece o valor E de correção no caso de utilização do banco de dados BLAST segmentado.

O NOBLAST utiliza o pacote Blastall da NCBI para o seu funcionamento. O Blastall possui os parâmetros **-m** 18 e **-m** 19 que produzem um longo texto no formato de arquivo tabular delimitado com todos os *hits* BLAST e seus respectivos atributos. O NOBLAST faz uso destas duas opções do Blastall na geração de seus resultados. A única diferença entre as duas opções é que **-m** 18 armazena cada alinhamento, ao passo que **-m** 19 não.

Este aplicativo é uma extensão para o conjunto de ferramentas da NCBI, que fornece um novo formato de saída tabular estendido para os programas BLAST (Blastn, Blastp, Blastx, Tblastn, Tblastx, MegaBLAST e BLAST Psi). Este formato pode ser diretamente analisado em qualquer planilha eletrônica ou qualquer outro mecanismo de banco de dados como o MySQL. O NOBLAST oferece uma opção para corrigir as estatísticas (correção de E-value), quando são utilizadas bases de dados divididas ou quando é necessária a normalização do BLAST.

Todos os recursos do padrão de saída do BLAST NCBI estão presentes, além de alguns outros, tais como: consulta e índice de conteúdo, consulta e cobertura do comprimento do conteúdo de um alinhamento, consultas independentes e o número de lacunas de um conteúdo, acertos de classificação para uma determinada consulta e pares de pontuação alta (*HSPs*) e classificação para um dado acerto. Estes índices podem ser úteis para remontar os resultados do BLAST quando são utilizadas consultas ou bases de dados divididas.

Embora, a saída produzida com a opção **-m** 18/19 possa ser carregada e analisada diretamente em qualquer planilha ou banco de dados, o JAMBLAST é uma alternativa criada em Java que pode ser usada opcionalmente para a gestão das saídas BLAST produzidas pelo NOBLAST, através do banco MySQL.

O JAMBLAST (LAGNEL; TSIGENOPOULOS; LLIOPOULOS, 2009), que utiliza a saída do NOBLAST para o seu funcionamento, é uma interface gráfica que permite ao usuário gerenciar, exibir e filtrar os acertos BLAST usando uma série de critérios de seleção. O JAMBLAST é escrito em JAVA, mas não é um aplicativo servidor. Este aplicativo necessita da instalação local do banco de dados MySQL e também do Java JRE.

O JAMBLAST também oferece visualização, classificação e filtragem dos resultados de acordo com critérios definidos pelo usuário. O JAMBLAST pode ser utilizado diretamente em qualquer computador pessoal, não necessitando de nenhum servidor exclusivo para tanto.

O primeiro quadro da Figura 2.6 apresenta cópias de tela das características e opções da tabela de saída BLAST através da aplicação JAVA JAMBLAST para visualização. Os resultados podem ser ordenados e filtrados por diferentes opções com base na escolha do usuário. O arquivo resultante poderá ser exportado em formato de texto delimitado tabular e cada alinhamento poderá ser visto após a seleção.

A cópia de tela, no segundo quadro da Figura 2.6, apresenta o principal filtro do JAMBLAST que pode ser usado para selecionar os alinhamentos a serem exportados. Mais especificamente, o usuário poderá filtrar por E-value, pelo percentual de cobertura de comprimento do alinhamento para consulta e/ou assunto e pelo percentual de identidade de sequências. Além disso, o usuário será capaz de definir um limite para o comprimento mínimo de alinhamento, manter o número de visitas que atendam suas necessidades e escolher ou remover os auto-acertos. Finalmente, o usuário poderá digitar uma expressão SQL padrão.

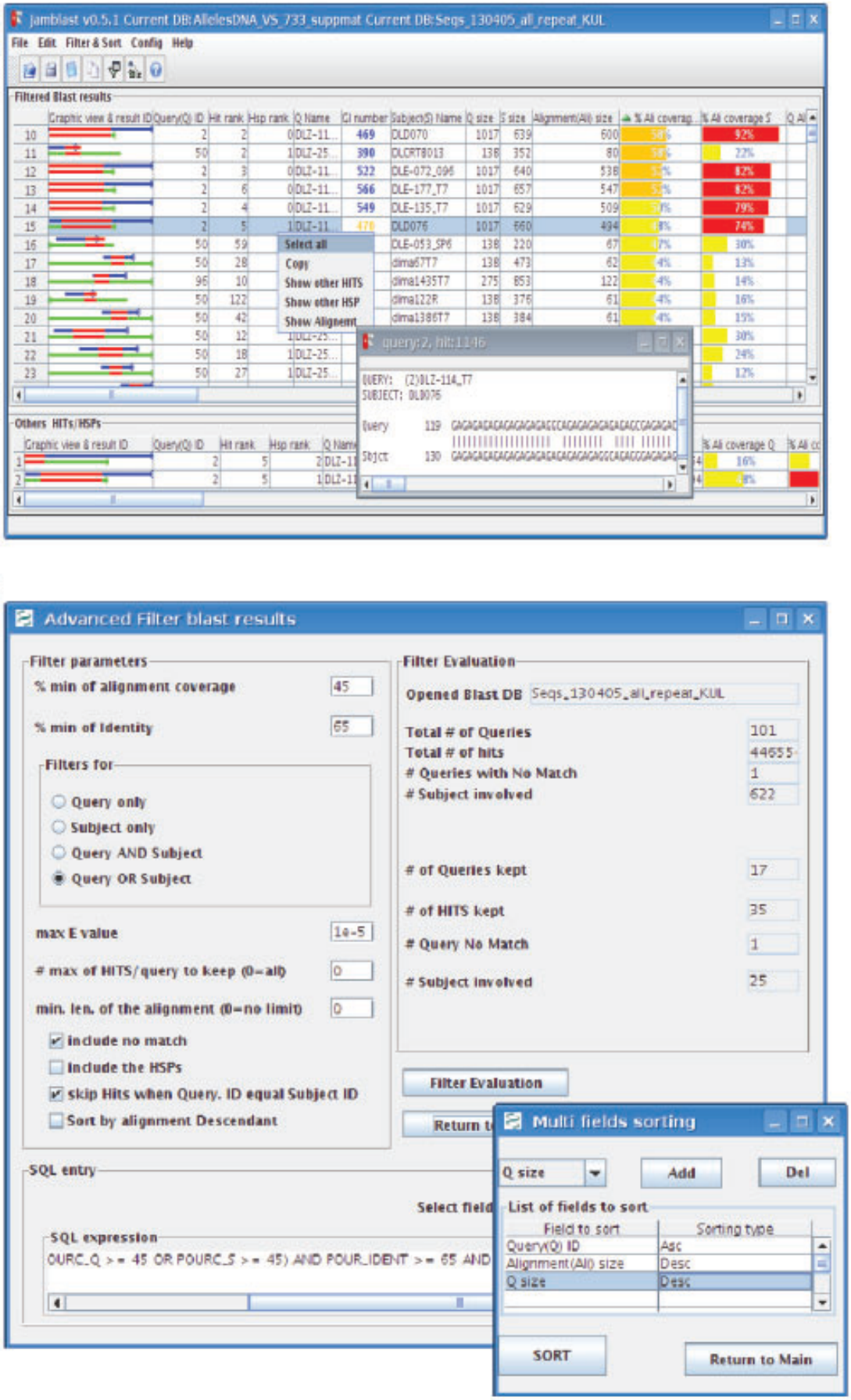


Figura 2.6: Cópias de tela da interface do JAMBLAST.

2.4 BlastQuest

O BlastQuest (FARMERIE et al., 2004) baseia-se na tecnologia de banco de dados e fornece consultas para a internet, análise e visualização de dados genômicos. A interface com a ontologia genética (GO) e as bases de dados fomentam o fluxo de trabalho biológico. Seguindo o mesmo padrão do NuclearBlast, o BlastQuest usa um conjunto de dados armazenado localmente, manualmente baixado do endereço *ftp* do NCBI.

O BlastQuest foi desenvolvido para a análise em projetos de sequenciamento de genes, fornecendo um conjunto de ferramentas integradas para análise genética. Mais especificamente, o BlastQuest permite filtrar, resumir, classificar e agrupar dados de buscas do BLAST, e complementar a saída com anotações de ontologia genética. Além disso, os dados do BLAST podem ser ligados à SMART (Ferramenta de Pesquisa Simples de Arquitetura Modular).

O BlastQuest usa XML como sua linguagem interna de representação de dados e armazenamento de todos os dados, incluindo o resultado de experiências, bem como o vocabulário GO, todos os KEGG ortólogos e suas relações com as vias em um banco de dados relacional. Isso permite o uso das características de bancos relacionais, como transações, compartilhamento controlado, e otimização da consulta.

O fato de os dados estarem armazenados localmente possibilita um bom desempenho e oferece flexibilidade para os usuários, porém não existe garantia de que os dados manipulados armazenados estejam atualizados. O BlastQuest também suporta um conjunto de funções de análise. As mais utilizadas são diretamente acessíveis através de botões de comando.

Para habilitar a análise de dados que não são diretamente suportadas pela interface padrão, o BlastQuest oferece uma interface de consulta não-SQL com baixo nível de complexidade. Esta interface permite ao usuário construir expressões booleanas complexas como condições de seleção, que incluem operadores lógicos e predicados de sub-caracteres de pesquisa.

A execução da consulta subjacente é baseada em consultas SQL parametrizadas que são instanciadas e automaticamente convertidas em código SQL executável pelo sistema de gerenciamento de banco de dados relacional. O BlastQuest também permite aos usuários gerenciar os dados BLAST do experimento relacionado em uma base por projeto ou por usuário, utilizando os recursos de segurança do RDBMS, enquanto que ao mesmo tempo permite o compartilhamento controlado de dados a fim de apoiar a colaboração.

O BlastQuest apresenta uma página de resumo para todas as sequências selecionadas ou de consulta. Para cada sequência de consulta, o acerto superior do BLAST (ou seja, a sequência do banco de dados de sequências com a melhor pontuação estatística) é, por padrão, considerada como a melhor sequência de correspondência. Ela é exibida com um resumo das informações biológicas importantes, que contém, para cada sequência, o ID de sequência (por exemplo, o número de GI do GenBank), a definição do gene e o valor esperado. O primeiro quadro da Figura 2.7 apresenta esta página de resumo.

Ao ser habilitada a opção de conversão *Amino conversion* e ao se clicar no botão *Details* novamente, ocorre uma tradução para uma sequência de aminoácidos, que é automaticamente submetida ao banco de dados SMART (SCHULTZ et al., 2000) para pesquisa de domínio, como mostrado no segundo quadro da Figura 2.7.

A Figura 2.8, em seu primeiro quadro, mostra como agrupar resultados BLAST baseados nos números de GI, para todas as sequências no projeto. Clicar em um acerto específico, fará com que seja exibida uma página contendo todas as sequências de consulta que coincidirem com esta sequência de banco de dados.

O segundo quadro da Figura 2.8 mostra 12 sequências de genes que codificam a consulta com a correspondência de regiões determinadas. Este é um método estabelecido para identificar sequências que vêm de diferentes regiões do mRNA, mesmo de genes ortólogos, ou parálogos. Este agrupamento também revela a transcrição de possíveis variantes.

Como mostrado no primeiro quadro da Figura 2.9, a Ontologia Genética possui três categorias principais, a saber, os processos biológicos, funções moleculares e componentes celulares, que estão subdivididos em subcategorias de várias profundidades. Grafos acíclicos dirigidos são formados por termos da Ontologia Genética e seus associados **é um** e **é parte de** nos relacionamentos.

A Figura 2.9, no segundo quadro, mostra os ramos de processos fisiológicos no gráfico de Ontologia Genética. Quando um usuário clica na opção *Contains* ou *Not Contains*, apresentada no primeiro quadro da Figura 2.10, uma janela é exibida, conforme o segundo quadro da Figura 2.10, onde o usuário poderá digitar uma cadeia de texto.

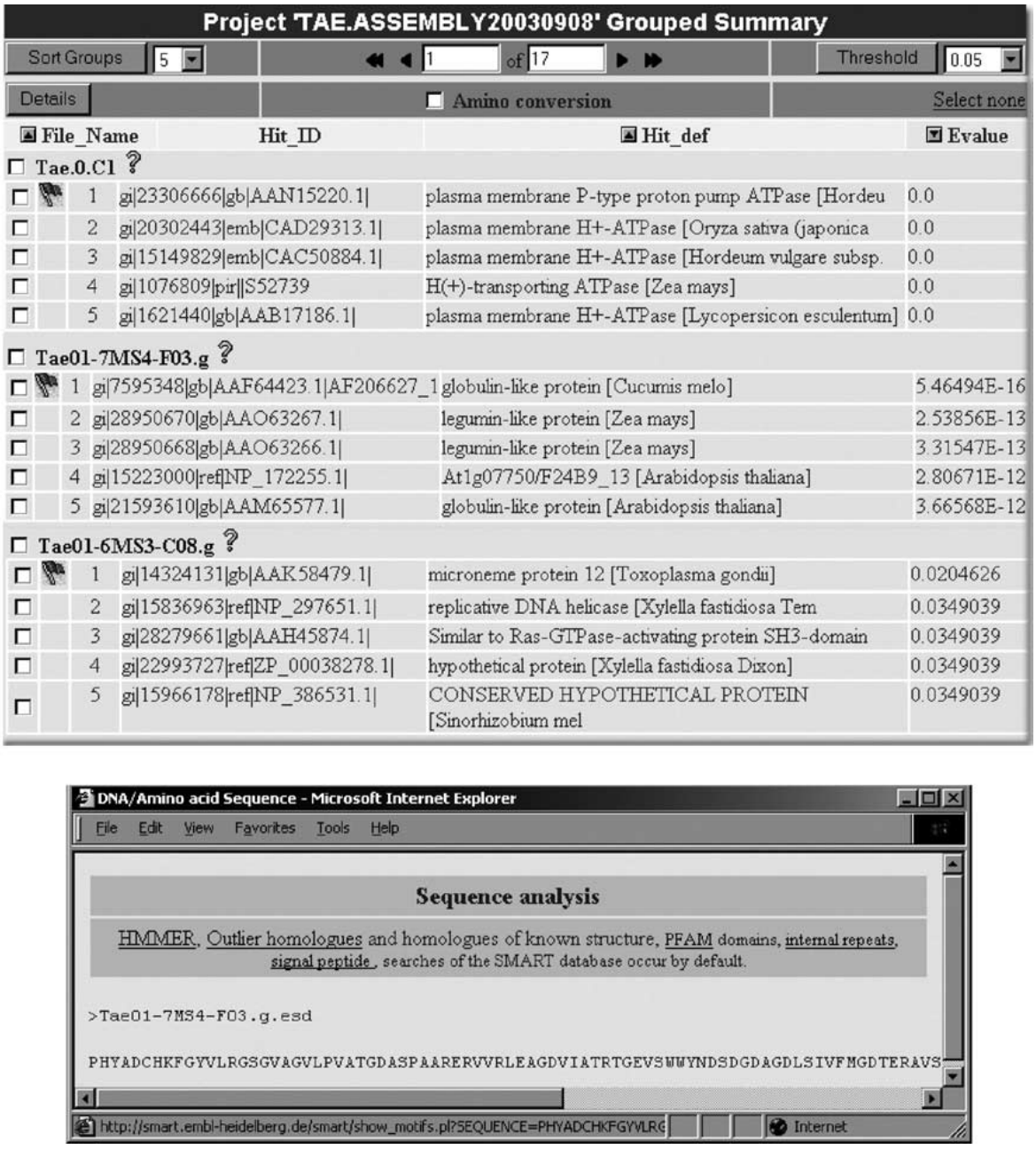


Figura 2.7: Checagem de consistência e filtragem dos resultados BLAST e interoperabilidade com outros sistemas através do BlastQuest.

# Hits	Hit_ID	Hit_def
12	gi|8918359|dbj|BAA97583.1|	RuBisCO activase large isoform precursor [Oryza sativa (japonica cultivar-group)]
9	gi|12643756|sp|Q40073|RCAA_HORVU	ribulose 1,5-bisphosphate carboxylase activase isoform 2 [Hordeum vulgare subsp. vulgare]
9	gi|100934|pir||S20925	polyubiquitin [Zea mays]
8	gi|1657859|gb|AAB18209.1|	chlorophyll a/b-binding protein WCAB precursor [Triticum aestivum]
8	gi|31753114|gb|AAH53854.1|	Unknown (protein for IMAGE:5194336) [Homo sapiens]
8	gi|23397122|gb|AAN31845.1|	putative polyubiquitin (UBQ10) [Arabidopsis thaliana]
8	gi|399414|sp|Q03033|EF1A_WHEAT	elongation factor 1-alpha [Hordeum vulgare subsp. vulgare]
8	gi|70644|pir||UQFS	ubiquitin precursor - common sunflower (fragment)
7	gi|5499713|gb|AAD43962.1|U78762_1	receptor-like kinase ARK1AS [Triticum aestivum]
7	gi|8918361|dbj|BAA97584.1|	RuBisCO activase small isoform precursor [Oryza sativa]
7	gi|10720253|sp|Q42450|RCAB_HORVU	ribulose 1,5-bisphosphate carboxylase activase [Hordeum vulgare subsp. vulgare]
7	gi|70645|pir||UQPM	1603402A poly-ubiquitin
6	gi|167096|gb|AAA63163.1|	ribulose 1,5-bisphosphate carboxylase activase isoform 1 [Hordeum vulgare subsp. vulgare]
6	gi|5523856|gb|AAD44031.1|	receptor-like kinase [Hordeum vulgare]

RuBisCO activase large isoform precursor [Oryza sativa (japonica cultivar-group)]

Select all | Select none

Fasta | Hit_Sequence_length = 466

		Query_def	QLen	QStart - QEnd	HStart	HEnd	HDiff	Q/H Frame	evalue
1	☐	Tae.5.C1	1430	183 - 1430	1 - 413		412	3 / 0	0
2	☐	Tae.5.C2	811	285 - 758	271 - 428		157	-3 / 0	0
3	☐	Tae.5.C3	752	140 - 751	1 - 201		200	2 / 0	0
4	☐	Tae.5.C4	1507	123 - 1403	1 - 428		427	3 / 0	0
5	☐	Tae01-1MS2-C07.g	708	2 - 658	247 - 466		219	2 / 0	0
6	☐	Tae01-2MS4-F11.g	683	176 - 682	1 - 166		165	2 / 0	0
7	☐	Tae01-3MS1-D06.g	718	187 - 717	42 - 214		172	1 / 0	0
8	☐	Tae01-4MS3-F05.g	688	67 - 588	1 - 175		174	1 / 0	0
9	☐	Tae01-7MS1-E12.g	712	197 - 712	1 - 169		168	2 / 0	0
10	☐	Tae01-4MS4-E05.g	371	127 - 327	70 - 137		67	1 / 0	0
11	☐	Tae01-7MS2-B10.g	402	188 - 301	428 - 466		38	2 / 0	3.12762e-25
12	☐	Tae01-2MS3-E11.g	499	196 - 264	1 - 23		22	1 / 0	2.78098e-10

Figura 2.8: Agrupamento dos resultados do BLAST em uma base de projeto.

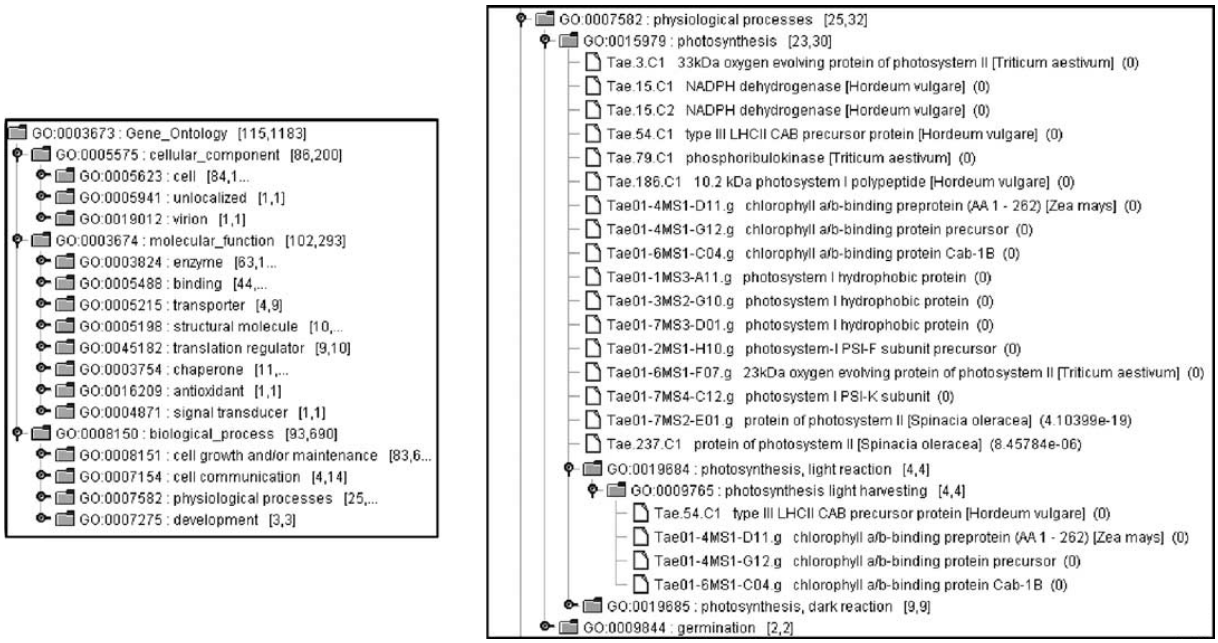


Figura 2.9: Funções de categorização baseada na ontologia genética.

Figura 2.10: Exemplo de consulta não-SQL do BlastQuest.

2.5 WebBlast

O WebBLAST (FERLANTI et al., 1999), como o próprio nome sugere, é uma aplicação servidora desenvolvida para a internet. O WebBLAST permite a manipulação de dados e análise elementar nos primeiros estágios de maturação dos dados da sequência.

Uma das principais características do WebBLAST é que seus requisitos de sistema são pequenos. O programa exige apenas o MacPerl no cliente Macintosh, Perl 5 e BLAST 2.0 no cliente UNIX e um servidor web.

Os dados do WebBLAST entram em um fila de espera e são executados em ordem. Depois que o processo de sequenciamento é concluído, o software de sequenciamento cria uma pasta contendo a sequência e os arquivos de rastreamento. Esta pasta recém-criada, ao ser arrastada para o processo chamado de *(SequenceUpload)*, faz com que seja iniciada a transferência. Os arquivos são então transferidos para um diretório armazenado no servidor UNIX.

Todos os dados são armazenados no servidor UNIX, em um banco de dados do sistema de arquivos sob o diretório raiz do servidor web. Tipicamente, o sequenciamento recebe o seu próprio projeto, e cada clone dentro desse sequenciamento é designado como um subprojeto. Cada diretório de sequências contém a sequência, o rastreamento para a sequência, e relatórios BLAST, entre outros dados. Um agendamento noturno invoca o processamento de novos dados previamente carregados. As sequências são testadas em um vetor, a ALU e outras sequências repetitivas são então mascaradas.

As buscas locais BLAST são em seguida executadas, os dados mascarados e os resultados são gravados então no banco de dados. Os arquivos de configuração permitem aos usuários especificar quais programas BLAST são chamados, quais bancos de dados são utilizados para as consultas BLAST e quaisquer opções de linha de comando que devam ser passadas para o BLAST.

Os relatórios resultantes do BLAST são analisados, e os pares com pontuação alta que satisfazem as estatísticas de corte são gravados em disco em formato tabular. Finalmente, as rotinas podem ser configuradas de forma que as coleções de dados são reprocessadas uma vez por mês usando o banco de dados de sequências do GenBank, no intuito de manter os resultados obtidos através do BLAST atualizados.

O acesso aos dados é feito através de uma interface para a internet, permitindo aos usuários, acesso aos dados independentemente da plataforma. Ao selecionar uma

coleção, um quadro resumido de acertos significativos para todas as sequências dentro dessa coleção é exibido. A partir da tabela resumida, os usuários podem selecionar uma sequência individual, que produz uma página de detalhes para a amostra.

Um miniaplicativo Java no topo da página de detalhes dá uma visão gráfica da distribuição de visitas BLAST, com cada acerto BLAST sendo representado por uma barra de cor. As informações complementares são mostradas abaixo da visualização gráfica, juntamente com um quadro-resumo para cada tipo de BLAST executado. Estes são, por sua vez, seguidos por tabelas classificadas, indicando os resultados individuais produzidos por cada execução do BLAST.

Ao se clicar sobre um número de registro contido dentro destas tabelas, é então exibido ao usuário os alinhamentos relevantes no relatório BLAST. Os números gerados dentro do relatório BLAST são *hyperlinks* para o NCBI Entrez (Baxevanis, 1998), que possibilitam a obtenção de informações adicionais sobre o ítem selecionado.

A Figura 2.11 apresenta visualizações do WebBLAST geradas pelo programa web-blast.cgi no primeiro quadro, além de uma página de resumo para um conjunto de dados individuais. Uma janela de comentários é mostrada na inserção.

Informações de comentários completos são visíveis em qualquer janela adicional exibida por um clique do mouse, ou na página de detalhes, como é apresentado no segundo quadro da Figura 2.11 (página Detalhe para uma sequência individual). O visualizador gráfico mostra todas as HSPs, enquanto a contagem de acertos reais mostrados nas tabelas representam o número absoluto de sequências encontradas. As colunas da tabela podem ser classificadas clicando-se em qualquer cabeçalho da coluna.

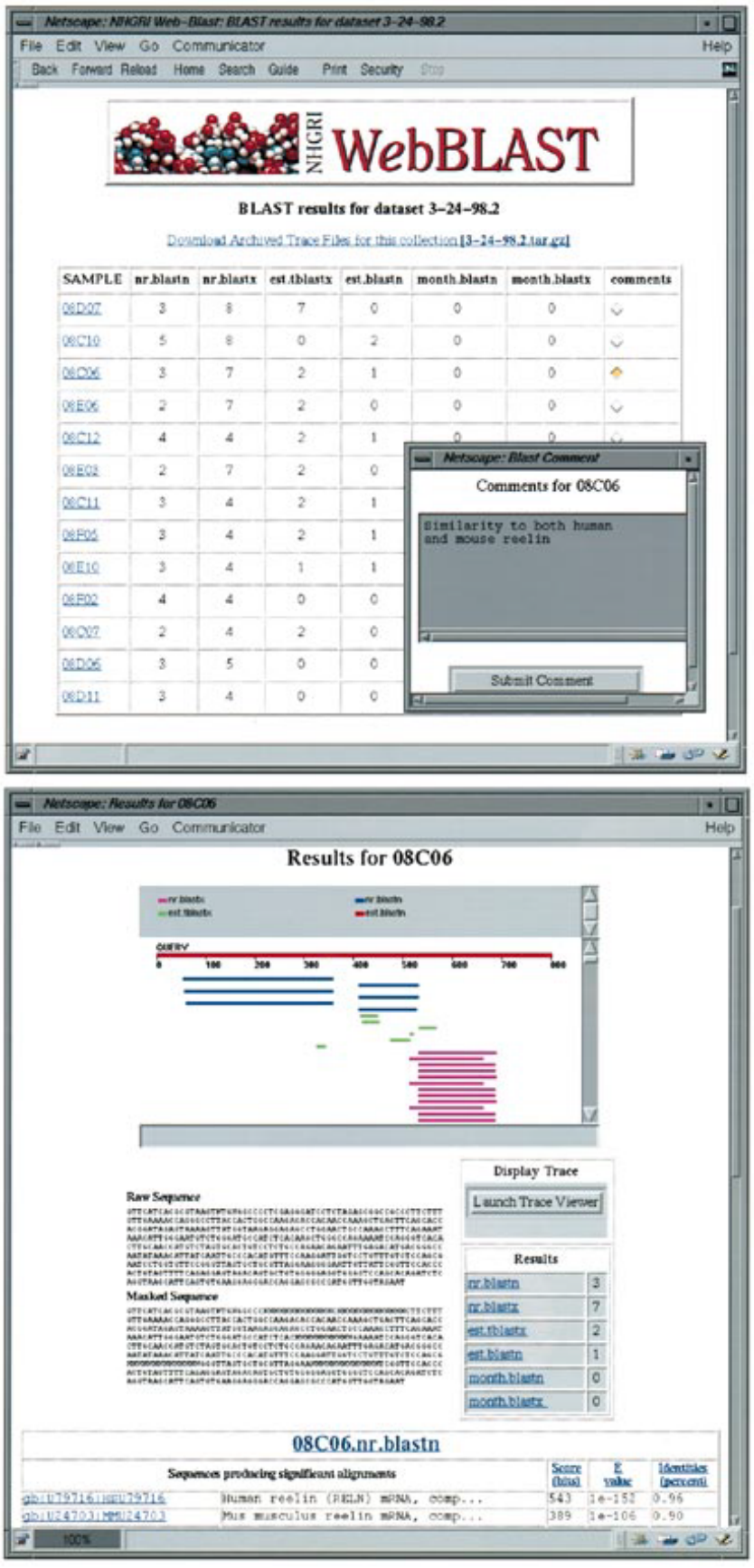


Figura 2.11: Cópias de tela do formulário principal do WebBlast.

2.6 ARB

O ARB (do latim arbor: árvore) (LUDWIG et al., 2004) é um projeto que foi criado como uma iniciativa interdisciplinar entre a Lehrstuhl für Mikrobiologie e a Lehrstuhl für Rechnertechnik und Rechnerorganisation, Parallelrechnerarchitektur, ambas unidades da Universidade Técnica de Munique em meados de 1994.

Nesta época, a análise da sequência comparativa das pequenas subunidades rRNAs ou dos genes respectivos já havia sido estabelecida como a abordagem mais comumente aplicada para a inferência filogenética, assim como a identificação de taxonomias microbianas (LUDWIG et al., 2004).

As técnicas de sequenciamento melhoradas e automatizadas promoveram um rápido aumento no número de pequenas subunidades rRNAs, também conhecidas como RNAs ribossomais, estruturas primárias de dados disponíveis a partir de fontes de dados como o GenBank ou EBI (European Bioinformatics Institute). No entanto, esses bancos de dados fornecem apenas dados brutos e informações descritivas adicionais que não podem ser estendidas de forma interativa pelo usuário.

Embora o projeto de banco ribossômico (RDP) (BL et al., 2001) e os projetos de Antwerpen (WUYTS et al., 2002) ofereçam conjuntos de dados de sequências alinhadas, a manipulação dos dados e a análise permaneceram difíceis para os cientistas que aplicam métodos baseados em rRNA.

Uma variedade de ferramentas de software individuais para edição de sequências, alinhamentos e análises filogenéticas estavam disponíveis em meados de 1994, oriundas de projetos de bancos de dados diferentes (STOESSER et al., 2002; BENSON et al., 2009; BL et al., 2001) e ainda de outras fontes adicionais.

Todavia, um pacote completo de ferramentas de interação estava faltando. Além disso, um grande número de diferentes formatos de entrada e saída tiveram que ser usados, oriundos de uma grande variedade de programas independentes. Estes dados precisaram passar por um processo de padronização para que se tornassem utilizáveis.

Infelizmente, uma iniciativa promissora, o projeto *Genetic Data Environment (GDE)* (LUDWIG et al., 2004), com foco no desenvolvimento de uma interface gráfica para manipulação de dados e análises foi descontinuado.

Microbiologistas e cientistas da computação da Universidade Técnica de Munique decidiram então desenvolver seu próprio pacote de software, que fosse capaz de gerir

corretamente os dados futuros. As duas principais tarefas, de acordo com o conceito ARB, formuladas no início do projeto e mantidos até o presente momento, são a manutenção de um banco de dados estruturado secundário, combinando estruturas primárias processadas e qualquer tipo de dados adicionais atribuídos por entradas de sequência individuais.

Além dos pontos mencionados, é possível incluir uma seleção abrangente de ferramentas de software que interagem diretamente umas com as outras, bem como um banco de dados central, sendo estes módulos controlados através de uma interface gráfica comum. As bases de dados de software e rRNA estão acessíveis ao público através do endereço http://www.arb-home.de.

O ARB foi desenvolvido para sistemas UNIX e seus derivados. Em suas últimas versões, o desenvolvimento foi realizado utilizando o SuSE LINUX (TEAM, 2011f) como sistema operacional. A maior parte do código-fonte foi escrito em C++ e C; algumas partes foram escritas em Perl dentre outras.

O ambiente gráfico é baseado na biblioteca Open Motif (TEAM, 2011c). As funcionalidades do projeto GDE sobre a edição de sequências foram adotadas e implementadas no pacote ARB. Alguns programas do pacote PHYLIP para inferência em filogenia (FELSENSTEIN, 1989) foram incorporados como componentes, interagindo diretamente com o banco de dados central.

Além disso, o fastDNAml (OLSEN et al., 1994) e o Protml do pacote Molphy (ADACHI; HASEGAWA, 1996), os componentes do pacote Puzzle (STRIMMER; HAESELER, 1996) e o AxML, um derivado do fastDNAml (STAMATAKIS et al., 2002), foram incluídos para máxima verossimilhança baseada em análises filogenéticas de ácidos nucleicos e aminoácidos, dados em sequência. O diagrama da Figura 2.12 apresenta esquematicamente as ferramentas do ARB e suas interações umas com as outras e com o banco de dados central.

A maioria das ferramentas desenvolvidas para o ARB interagem diretamente com uma cópia do banco de dados na memória principal, enquanto que a segunda parte integra ferramentas que são fornecidas com os dados do ARB e seus resultados são escritos novamente no banco de dados. Assim, quaisquer alterações ou rearranjos são imediatamente replicados para os componentes de software periféricos.

A Figura 2.13 mostra que o resultado total da consulta ou apenas uma seleção feita pelo usuário no banco de dados poderá ser visualizada nas janelas respectivas. Nesta figura, os dados bibliográficos armazenados em banco de dados e os respectivos campos são mostrados, a seleção de campos do banco de dados, extração de dados e o leiaute da

janela de visualização podem ser personalizados pelo usuário.

Árvores filogenéticas intrinsecamente geradas pela ferramenta ARB de reconstrução de árvores, ou importadas a partir de fontes externas são armazenadas no banco de dados e podem ser visualizadas em diferentes formatos dentro da janela principal do ARB, como mostra a Figura 2.14. Nesta figura, os retângulos representam grupos monofiléticos comprimidos *on-line* que podem ser desenvolvidos pelo clique do mouse. O Banco de dados de entradas Eld como nome taxonômico, número de acessos públicos do banco de dados e designação de tensão conforme relatado no EMBL, RDP e as bases de dados europeias rRNA (DEW) são visualizados nos nós do terminal desdobrado *Desulfohalobiaceae*.

Os dados da sequência podem ser visualizados e modificados por um editor, conforme apresenta a Figura 2.15. Como um exemplo, para ressaltar uma sequência de pesquisa de um site-alvo da sonda, é exibido um destaque através da cor de fundo. Emparelhamentos perfeitos ou incompatíveis são codificados também por cores. Os dados originais, bem como dados praticamente transformados (por exemplo: a *purina pirimidina* ou a apresentação de aminoácidos simplificados) são exibidos ao usuário através de códigos de cores.

O editor ARB de estrutura secundária apresentado na Figura 2.16 é válido para qualquer sequência no modelo de consenso comum. A sequência particular a ser visualizada é selecionada pelo posicionamento do cursor no editor de estrutura primária. A sequência selecionada no editor de estrutura primária é automaticamente enquadrada em um modelo de consenso de estrutura secundária.

Os alinhamentos locais são determinados entre a sequência alvo de sondagem e as sequências de referência mais similares (opcionalmente, de nenhum a cinco desencontros) na respectiva base de dados (Figura 2.17). Além disso, estas sequências podem ser automaticamente visualizadas nos editores de estrutura primária e secundária. Parte do alinhamento da estrutura primária que contém o endereço de destino da sonda é mostrado para o *retbaense Desulfohalobium*, organismo alvo, e os organismos não-alvo que contenham os trechos de sequência mais similares.

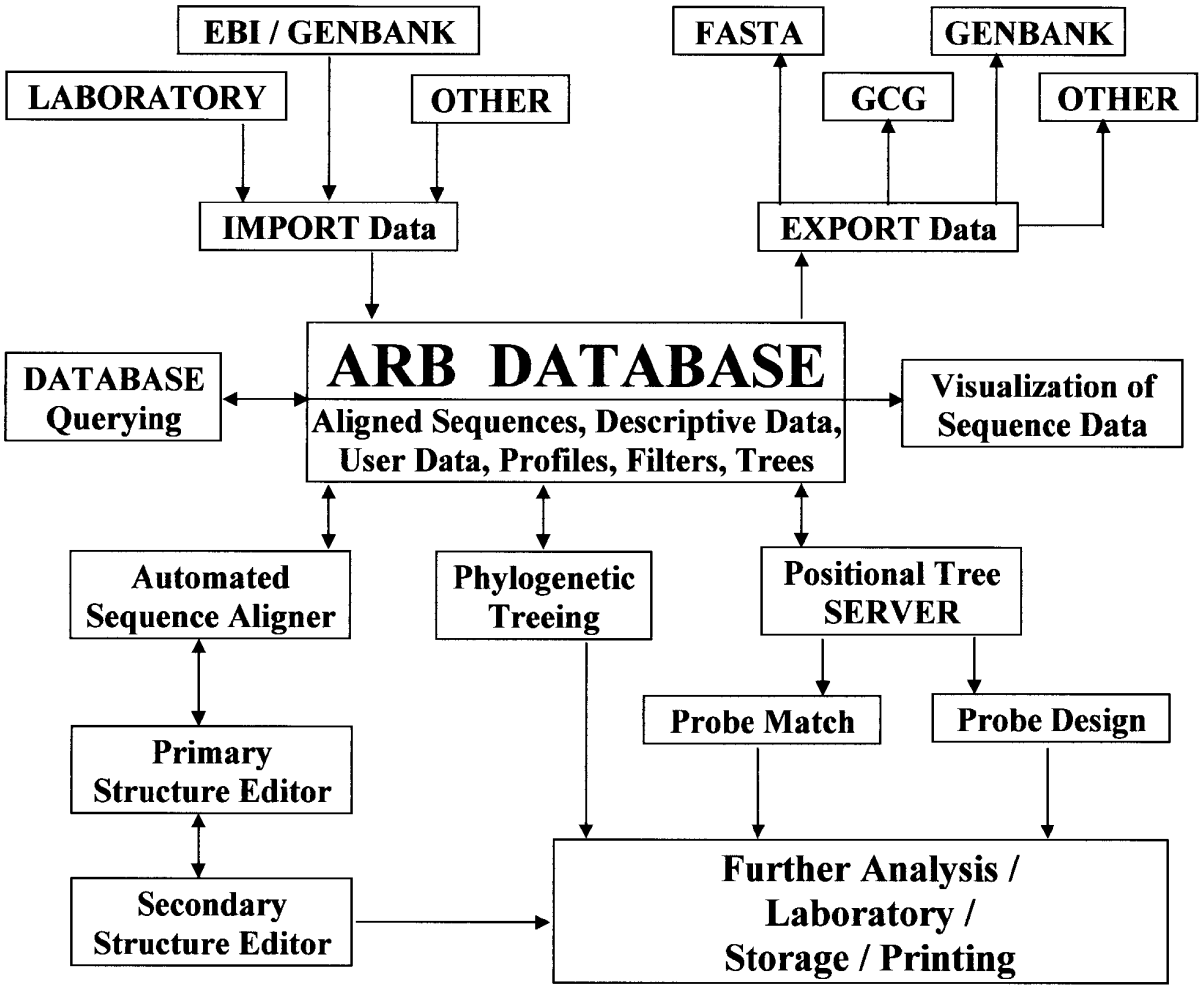


Figura 2.12: Interação entre componentes, ferramentas e o banco de dados no software ARB.

O ARB não utiliza o BLAST como seu algoritmo principal para alinhamento de sequências, porém o ARB possui uma interessante interface gráfica e vários recursos de análise de sequências, tornando-se relevante sua referência. De igual modo é possível fazer referência a outros projetos como o BLAST XS (MOON; LEFKOWITZ, 2011), OCGC (AL., 2011) e Pedante (FRISHMAN, 2001) que não serão apresentados em maiores detalhes neste trabalho.

O principal foco da maior parte dos projetos mencionados neste trabalho é o pós-processamento, ou seja, manipulações executadas sobre os resultados gerados pelo BLAST e seu conjunto de ferramentas, utilizando como entrada os resultados do próprio BLAST. Mas de maneira geral, estas ferramentas apresentam poucos parâmetros para a etapa da pesquisa e alinhamento das sequências, e a maior parte do processamento é executada após o retorno dos resultados do BLAST.

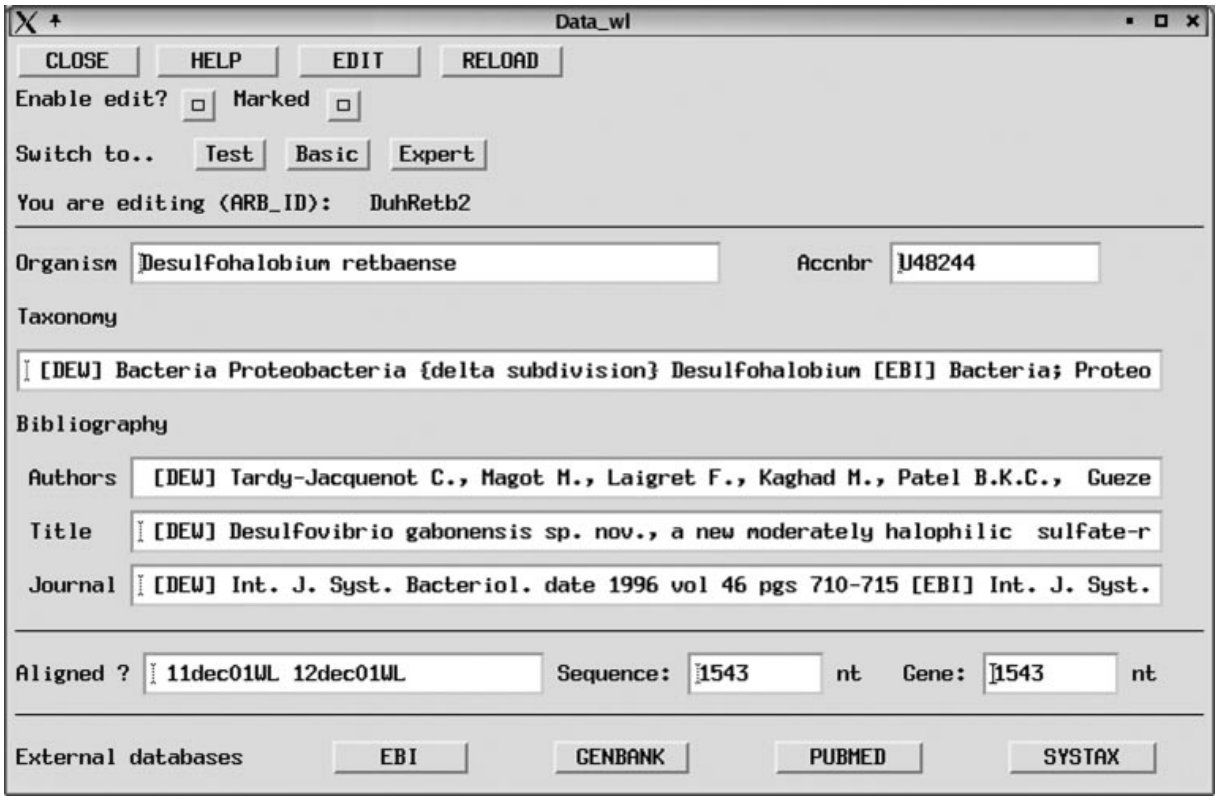


Figura 2.13: Exemplo de uma janela de visualização de dados.

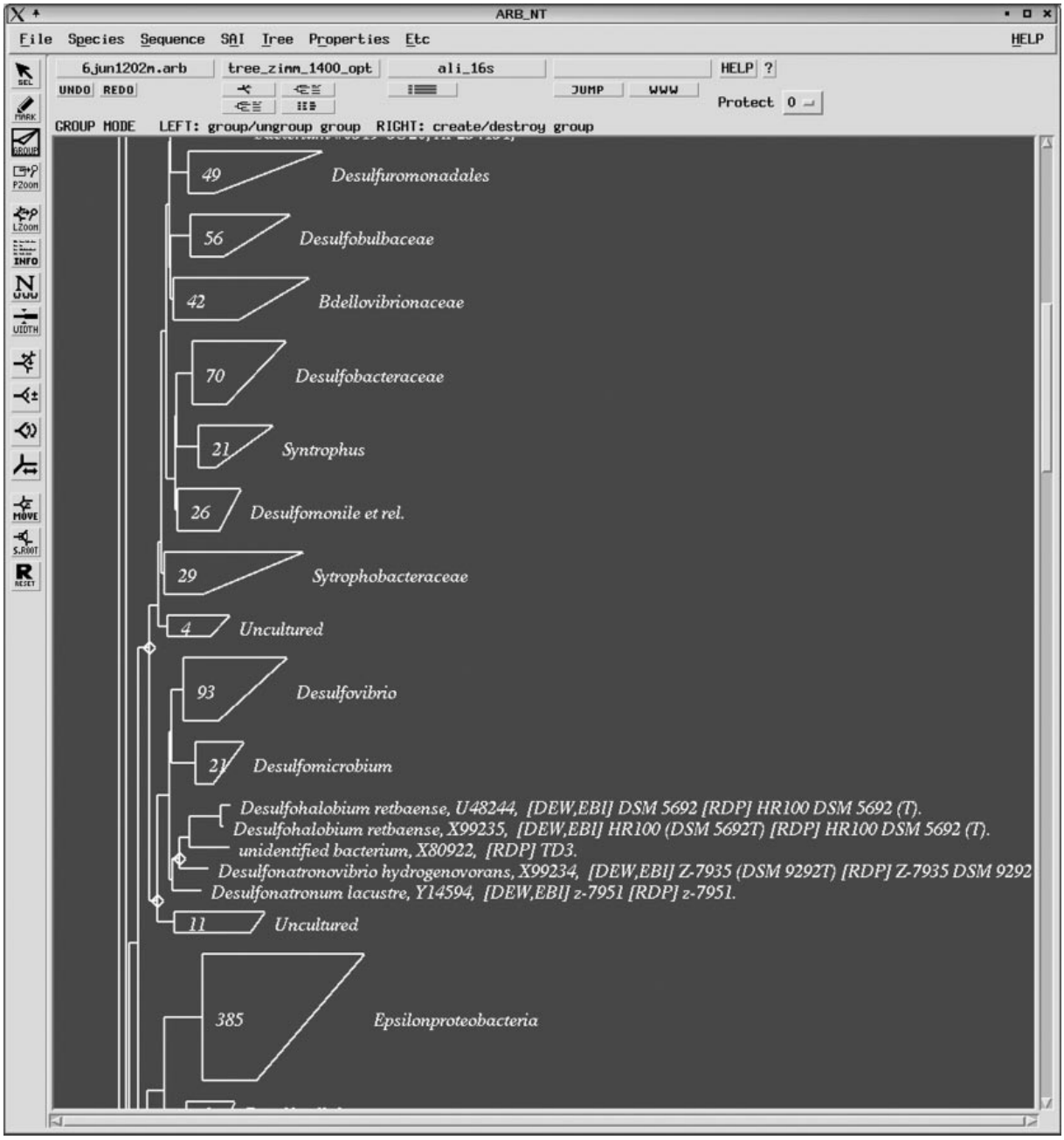


Figura 2.14: A janela principal, mostrando parte de um dendrograma gerado pela parcimônia ARB.

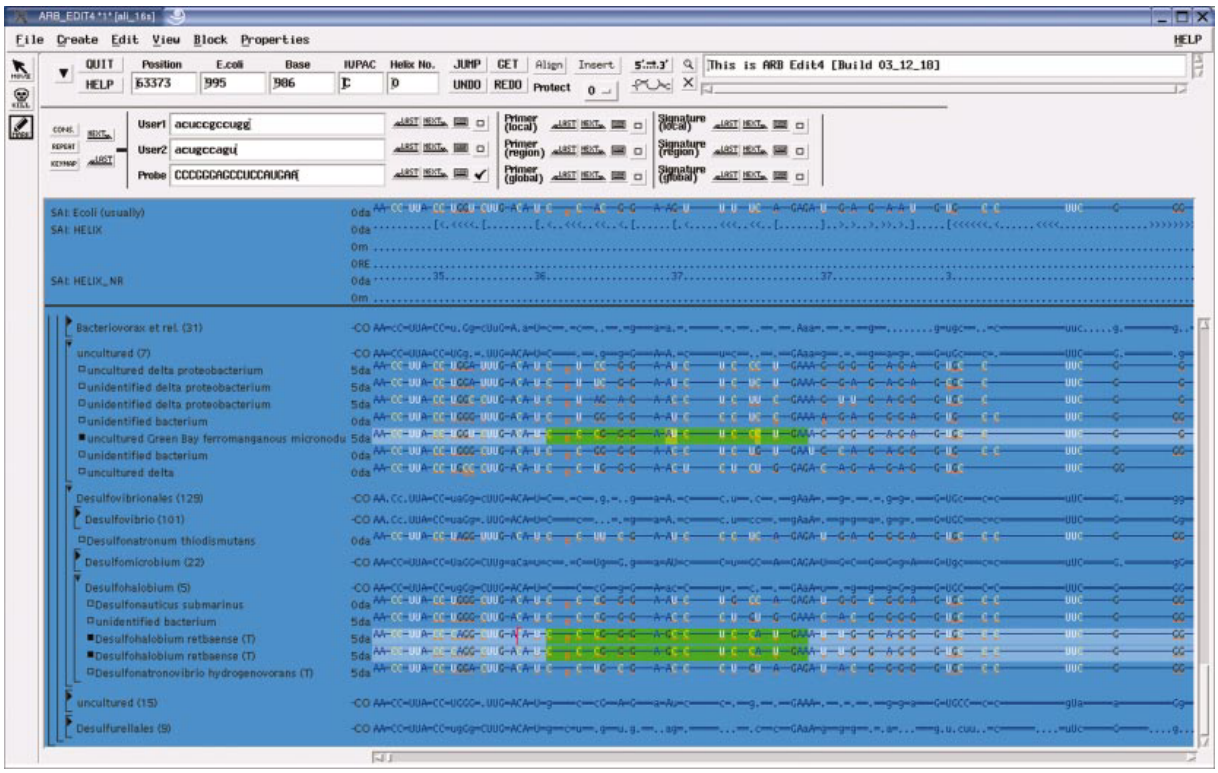


Figura 2.15: O ARB editor de estrutura primária.

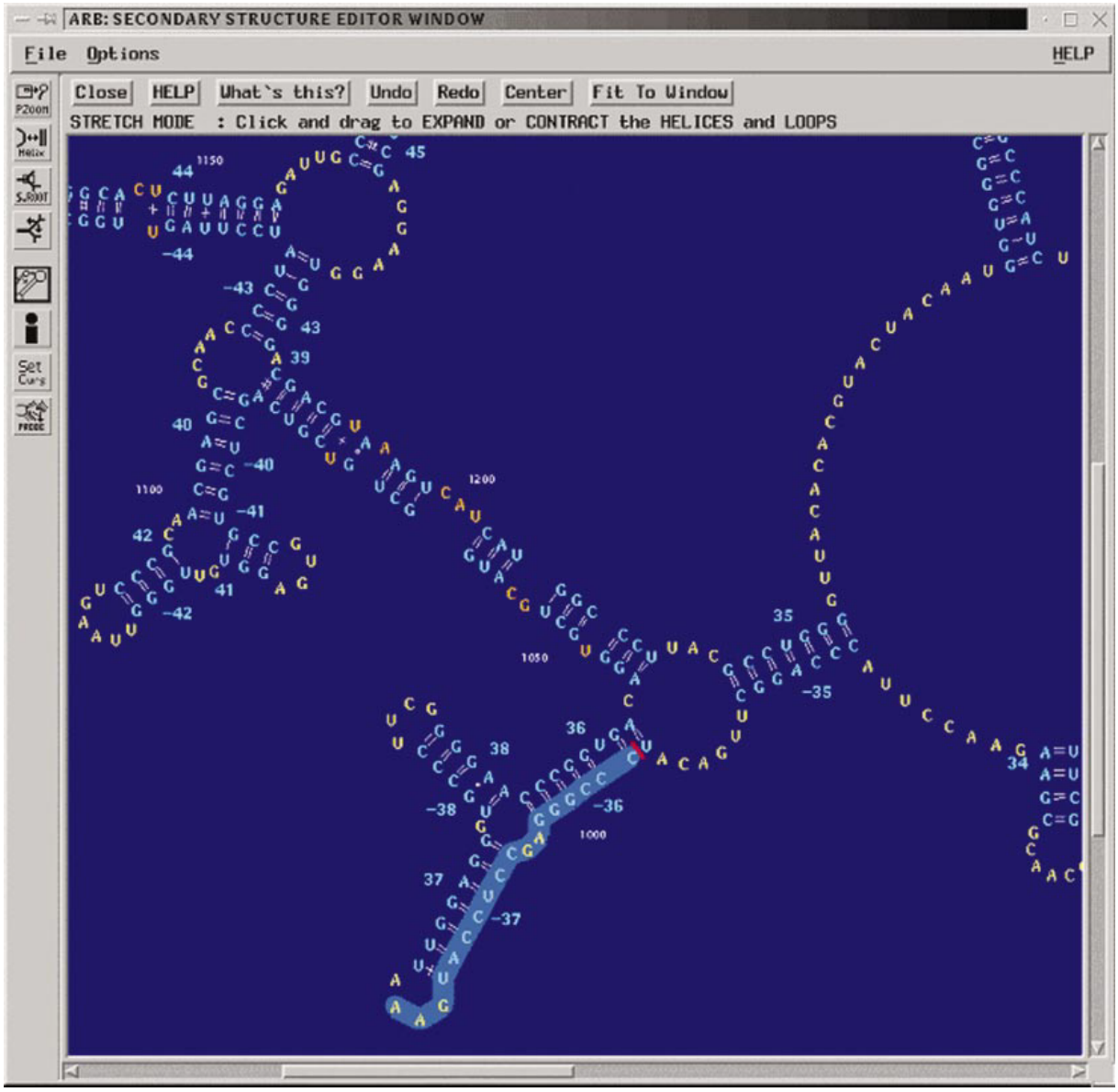


Figura 2.16: Editor de estrutura secundária.

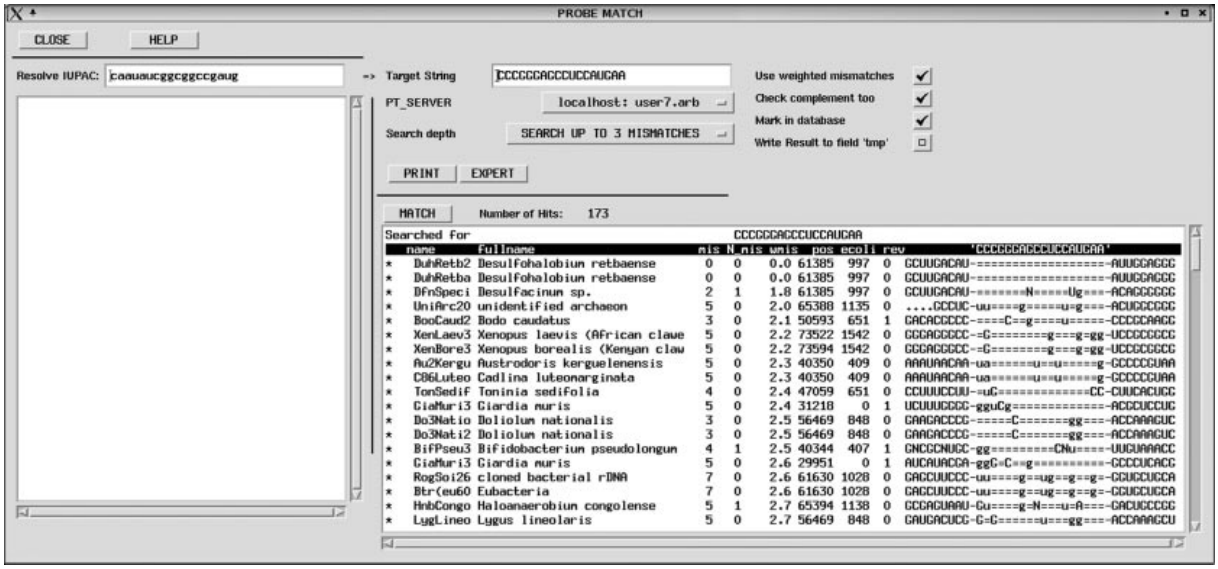


Figura 2.17: Resultados do projeto da sonda e avaliação.

3 Alinhamento de Sequências Através do Algoritmo BLAST

3.1 O Algoritmo BLAST

O BLAST - *Basic Local Alignment Search Tool* (ALTSCHUL et al., 1990) é um algoritmo de alinhamento local de sequências genéticas que efetua uma medição baseada em uma pontuação de mutações.

O BLAST aproxima diretamente os resultados que poderiam ser obtidos através de um algoritmo de programação dinâmica para otimizar esta medição. Este método irá detectar fracas similaridades entre sequências mas que, biologicamente falando, são muito significantes. Este algoritmo já apresentava desempenho superior em relação aos algoritmos heurísticos existentes na época de seu surgimento.

As medições de similaridade entre sequências podem ser classificadas como globais e locais. Os algoritmos de similaridade global otimizam a totalidade do alinhamento de duas sequências, porém isto pode fazer com que sejam incluidos no resultado grandes blocos que possuem pequena similaridade (WUNSCH, 1970). Já os algoritmos de similaridade local buscam somente subsequências relativamente conservadas e uma comparação simples poderia descartar vários alinhamentos subsequentes distintos.

O BLAST baseia-se no princípio de que regiões não conservadas não contribuem para a medição de similaridade (WATERMAN, 1981). Medições de similaridade local são geralmente preferíveis para buscas em bancos de dados, onde os cDNAs podem ser comparados com os genes parcialmente sequenciados, e onde proteínas distantemente relacionadas poderão compartilhar apenas isoladas regiões de similaridade.

Muitas medições de similaridade, incluindo as medições do BLAST, começam com a pontuação de uma matriz de similaridade para todos os possíveis pares de resíduos. Identidades e substituições conservativas possuem pontuação positiva, enquanto que substi-

tuições indesejáveis recebem pontuação negativa.

Para comparações de sequências de aminoácidos, o BLAST usa a matriz PAM-120 (uma variação da matriz de (DAYHOFF, 1978)), enquanto que para comparações de sequências de DNA a pontuação de identidade recebe o valor +5, e o valor -4 para diferenças, porém outras pontuações também são possíveis.

Um segmento de sequência é um trecho contínuo de resíduos de qualquer tamanho, e a pontuação de similaridade para dois segmentos alinhados do mesmo tamanho são a soma dos valores de similaridade para cada par de resíduos alinhados.

Dadas estas regras o BLAST define o par de segmentação máxima (*MSP*) para ser o par de maior pontuação de segmentos de tamanho idêntico, escolhidos à partir de duas sequências. Os limites de um *MSP* são escolhidos para maximizar sua pontuação, portanto um *MSP* pode ter qualquer tamanho. O *MSP* que o BLAST heuristicamente tenta calcular, provê a medição da similaridade local para qualquer par de sequências.

Porém, um biólogo molecular poderia estar interessado em todas as regiões conservadas que são compartilhadas por duas proteínas, e não apenas no seu par de maior pontuação. O BLAST deste modo define um par segmentado para ser localmente máximo se sua pontuação não pode ser melhorada nem por alongamento nem por encurtamento de ambos os segmentos (SELLERS, 1978).

O BLAST pode procurar todos os pares de segmentos locais com a pontuação acima de algum valor específico de corte. Assim como muitas outras medições de similaridade, a pontuação do *MSP* para duas sequências devem ser computadas em tempo proporcional ao produto de seus tamanhos usando um algoritmo simples de programação dinâmica.

Uma importante vantagem da medição do *MSP* é que os resultados matemáticos possibilitam a significância estatística da pontuação do *MSP* a ser estimado sob um modelo de sequência randômico apropriado (ALTSCHUL, 1990). Além disso, para qualquer matriz de pontuação em particular é possível estimar a frequência de resíduos pareados em segmentos máximos. Esta tratabilidade da análise matemática é um recurso crucial para o algoritmo BLAST. A Figura 3.1 apresenta a logomarca da NCBI, também usada pelo BLAST.

3.1.1 A Rápida Aproximação da Pontuação MSP

Na busca de um banco de dados de milhares de sequências, geralmente apenas uma pequena parte, se houver, será homóloga à sequência da consulta. Os pesquisadores

Figura 3.1: Logomarca ofical da NCBI também usada pelo BLAST.

desse modo estão interessados na identificação apenas daquelas entradas de sequência de pontuação *MSP* acima de algum valor de corte com pontuação S.

Estas sequências incluem aquelas que compartilham alta similaridade significante com a consulta, assim como as sequências que se aproximam dessa pontuação. Este último conjunto de sequências deve incluir os acertos aleatórios de alta pontuação assim como as sequências mais distantes relatadas na consulta.

A significância biológica das sequências de alta pontuação devem ser inferidas quase que somente na base da pontuação de similaridade, enquanto que o contexto biológico das sequências com pontuação intermediária podem ser úteis na distinção de relacionamentos interessantes, do ponto de vista biológico.

Alguns resultados (ALTSCHUL, 1990) possibilitaram a estimativa da mais alta pontuação *MSP*, na qual as chances da existência de similaridade são grandes. Para acelerar as buscas em bancos de dados, o BLAST minimizou o tempo gasto em regiões de sequência cuja similaridade com a consulta tenha uma pequena chance de exceder esta pontuação.

O objetivo do BLAST é deixar um par de palavras serem um par segmentado com comprimento e tamanho fixo w. Com este intúito, a principal estratégia do BLAST é a de buscar apenas pares segmentados que contenham um par de palavras com uma pontuação de pelo menos T.

No escaneamento através de uma sequência, é possível determinar rapidamente quando ela contém uma palavra de tamanho w que possa parear com a sequência de consulta para produzir um par de palavras com pontuação maior que, ou igual ao limiar T.

Qualquer acerto desse tipo é estendido para determinar se ele está contido dentro de um par de segmentos cuja pontuação é maior, ou igual a S. Para as sequências abaixo do limiar T, existe uma grande chance de haver um par segmentado com uma pontuação de

pelo menos T.

Um pequeno valor para T aumenta o número de acertos e consequentemente o tempo de execução do algoritmo. Simulações randômicas permitiram a seleção de um limiar T que balanceasse estas considerações (ALTSCHUL et al., 1990).

3.1.2 Implementação do Algoritmo BLAST

A implementação do algoritmo BLAST ocorreu em 3 etapas: a compilação nominal de uma lista de palavras com alta pontuação; o escaneamento do banco por acertos; e a extensão dos acertos. Variações ocorrem quando o banco contém proteínas ou sequências de DNA.

Para proteínas, a lista consiste de todas as palavras com pontuação de pelo menos T quando comparada com alguma palavra na sequência de consulta. Deste modo, uma palavra da consulta será representada por nenhuma palavra na lista ou por muitas.

Para valores de w e T que foram encontrados como mais adequados, existem tipicamente a ordem de 50 palavras na lista para cada resíduo na sequência de consulta: Por exemplo, 12.500 palavras para uma sequência de tamanho 250. Com um pouco de cuidado na programação, a lista de palavras pode ser gerada em tempo essencialmente proporcional ao tamanho da lista.

A fase de escaneamento sofreu um problema clássico de algoritmos: a busca de uma sequência longa para todas as ocorrências de uma certa sequência curta. Supondo que $w = 4$ e mapeando cada palavra com um inteiro entre 1 e 20^4, então uma palavra pode ser usada como um índice dentro de uma matriz de tamanho $20^4 = 160.000$. Deixa-se a i-ésima entrada do item de uma matriz apontar para a lista de todas as ocorrências em uma sequência de consulta da i-ésima palavra.

Desta forma, enquanto o banco de dados é escaneado, cada palavra do banco de dados aponta imediatamente para o correspondente acerto. Tipicamente, apenas alguns milhares das 20^4 possíveis palavras estarão nesta tabela, e será fácil de modificar a abordagem para usar muito menos que os 20^4 ponteiros.

A segunda abordagem explorada para a fase de escaneamento foi o uso de um autômato finito determinístico ou máquina de estados finitos (ULLMAN, 1979). Um importante recurso para a construção do algoritmo BLAST foi a detecção do sinal nas transições, ao invés da detecção nos estados.

A utilização dos autômatos salvou um fator em espaço e tempo proporcional ao tamanho do alfabeto subjacente. Este método rendeu ao programa uma maior velocidade e ficou definida então a preferência dessa abordagem para uso geral. Com os tamanhos de consultas e configurações de parâmetros padrão, o BLAST escaneou um banco de dados de proteínas em aproximadamente 500.000 resíduos/segundo.

O processo de estender um acerto para encontrar o par segmentado máximo local contendo este acerto é feito de forma sequencial do início para o fim. Para economizar tempo, o processo de extensão em uma direção foi encerrado quando o par segmentado cuja pontuação ficava a certa distância abaixo da melhor pontuação encontrada para extensões mais curtas.

Para DNAs, foi utilizada uma lista de palavras mais simples. Desta forma, uma sequência de consulta de tamanho n rendeu uma lista de $n-w+1$ palavras, e novamente houveram comumente alguns milhares de palavras na lista. Nestes casos, é vantajoso comprimir um banco de dados, através da compactação de 4 nucleotídeos em um único *byte*, usando uma tabela auxiliar para delimitar as bordas entre sequências adjacentes.

Assumindo que $w >= 11$, por exemplo, cada acerto deve conter um acerto 8-mer, que cabe nos limites de um byte. Esta observação permitiu o escaneamento do banco de dados no modo *byte-wise*, ou seja, o modo de escaneamento possível apenas em um banco de dados em que os nucleotídeos estão compactados (4x1), e desta forma aumentou-se o desempenho do processo em 4 vezes.

Executando no modo SUN4 (ALTSCHUL et al., 1990), com uma consulta de comprimento típico, o BLAST escaneou aproximadamente 2 x 10^6 bases/segundo. Para pesquisadores que executam muitas buscas deste tipo por dia, carregar este banco de dados comprimido para a memória uma vez em um esquema de memória compartilhada resulta em uma substancial redução no tempo das consultas.

É possível notar que as sequências de DNA são altamente não-randômicas, com uma composição de base localmente tendenciosa e com elementos de sequência repetidos. Isto tem importantes consequências para o desenvolvimento de uma ferramenta de busca em bancos de dados de DNA.

Se uma dada sequência de consulta tem, por exemplo, uma subsequência $A+T-rich$, ou um elemento repetitivo ocorrendo frequentemente, então a busca no banco resultará em uma copiosa saída de resultados de pouco interesse. No caso do BLAST, estes dois problemas foram sanados com algumas pequenas mudanças.

O programa que produz a versão comprimida do banco de dados de DNA tabula as frequências de todas as 8 tuplas. Isto ocorre muito mais frequentemente do que o esperado por acaso, e são armazenadas e usadas para filtrar palavras não-informativas da lista de palavras da consulta.

Também, precedendo buscas completas em bancos de dados, a busca em uma sub-biblioteca de elementos repetitivos é executada, e a localização na consulta de acertos significantes é armazenada. Palavras geradas por estas regiões são removidas da lista de palavras da consulta para a busca completa. Acertos para a sub-biblioteca contudo, são informadas no resultado final. Estes 2 filtros permitem o alinhamento de regiões com composição tendenciosa, ou para regiões com elementos repetitivos para serem relatados, assim como regiões adjacentes não contendo estes recursos compartilham significante similaridade com a sequência de consulta.

A estratégia do BLAST permite numerosas variações. Uma versão do BLAST foi implementada utilizando a programação dinâmica para estender acertos, assim como permitir lacunas nos alinhamentos resultantes, porém isso retarda amplamente a velocidade da extensão.

Enquanto a sensibilidade das buscas de aminoácidos foi melhorada em alguns casos, a seletividade foi proporcionalmente reduzida. Dado o conflito de escolha entre a velocidade e seletividade para a sensibilidade, é questionável quando a versão com *gap* do BLAST constitui um avanço.

3.1.3 NETBLAST - O BLAST para a WEB

O Netblast foi desenvolvido como uma versão cliente do algoritmo BLAST que pode ser executada através da internet, efetuando suas buscas e alinhamentos diretamente no servidor do GenBank através de serviços da internet. Deste modo, o Netblast não necessita que os arquivos de dados de sequências do GenBank estejam salvos em um servidor local para que seja possível a execução de alinhamentos.

O Netblast efetua buscas por sequências similares à uma sequências de consulta. A consulta e o banco de dados pesquisados podem ser tanto os de peptídeos quanto os de ácidos nucleicos em qualquer combinação. O Netblast pode pesquisar apenas os bancos de dados mantidos no Centro Nacional de Informações sobre Biotecnologia (NCBI), em Bethesda, Maryland, EUA.

O Netblast é muito semelhante ao BLAST, mas enquanto o BLAST procura

por sequências em bancos de dados locais usando os recursos do um servidor local, o Netblast realiza apenas pesquisas remotas, diretamente no GenBank.

O Netblast suporta cinco diferentes programas da família BLAST. O primeiro é o BLASTP, que efetua buscas em um um banco de dados de proteínas. No BLASTP, cada sequência de banco de dados é comparada com a consulta em uma comparação de pares separados, proteína-a-proteína.

O segundo é o BLASTX, que efetua buscas por nucleotídeos, mas em um banco de dados de proteínas. Neste caso a consulta é traduzida, e cada um dos seis produtos é comparado com cada sequência do banco de dados, em uma comparação de pares separados, proteína-a-proteína.

O terceiro é o BLASTN, que efetua buscas em um banco de dados de nucleotídeos. No BLASTN, cada sequência do banco de dados é comparada com a consulta em uma comparação de pares separados nucleotídeo-a-nucleotídeo.

O quarto programa é o TBLASTN, que pesquisa por proteínas em um banco de dados de nucleotídeos. No TBLASTN, cada sequência de nucleotídeos do banco de dados é traduzida, e cada um dos seis produtos é comparado com a consulta em uma comparação de pares separados proteína-a-proteína.

Por fim o TBLASTX efetua buscas por nucleotídeos em um banco de dados também de nucleotídeos. No TBLASTX, a consulta e cada sequência do banco de dados são traduzidas em seis quadros, e cada um dos 12 produtos é comparado em 36 diferentes comparações no modo *pairwise*. O fato de que este programa envolve mais recursos de computação do que os outros, faz com que o servidor BLAST no NCBI não aceite pedidos de buscas no banco de dados não-redundante (nr).

Normalmente, o Netblast decide qual o programa BLAST será necessário, simplesmente ao verificar o tipo (proteína ou ácido nucleico) da sequência de consulta submetida e o banco de dados selecionado. No caso de buscas nucleotídeo-nucleotídeo, existem dois programas que podem fazer a pesquisa. Por padrão, o BLASTN é usado.

Para pesquisas usando o TBLASTX em vez disso, deve-se usar o parâmetro -TBLASTX. O Netblast só pode pesquisar bancos de dados remotos mantidos pela NCBI, no entanto, como se trata da maior rede de bancos de dados genéticos da atualidade, não serão apresentadas muitas dificuldades na busca de sequências genéticas, dado o volume de informações armazenadas. As pesquisas remotas exigem quase que nenhum recurso do computador cliente solicitante, reduzindo drasticamente os pré-requisitos necessários para uma busca e

alinhamento.

Um ponto de grande importância a ser mencionado é que os bancos de dados localizados no NCBI são atualizados diariamente. O uso do Netblast só não é recomendado quando os dados de sequência são confidenciais, devido ao tráfego destas informações através da internet e aos riscos existentes durante a transferência destas informações.

O BLAST é um método de pesquisa conduzida estatisticamente, que encontra regiões de semelhança entre a consulta submetida e as sequências do banco de dados, e produz o alinhamento permitindo lacunas nestas regiões. Dentro destas regiões alinhadas, a soma dos valores da matriz de pontuação de seus pares-símbolo constituintes é maior do que em algum nível isso pudesse ocorrer por acaso.

O Netblast requer a definição de um nível de expectativa para a pesquisa inteira. Por padrão, esse nível é de 10.0, o que significa que serão relatadas apenas as sequências que tiverem atingido uma pontuação que seria esperada de ocorrer ao acaso não maior do que 10 vezes nessa busca em particular.

3.1.4 NETBLAST - Interpretando a Saída do Netblast

Cada par segmentado alinhado tem uma pontuação normalizada expressa em *bits*, que permite estimar a magnitude do espaço de busca e através de exaustivos testes chegou-se a uma pontuação HSP ótima para casos genéricos (ALTSCHUL et al., 1997). Se a pontuação é de 30 *bits*, será necessário marcar, em média, cerca de 1 bilhão de pares de segmentos independentes (2^{30}) para se encontrar um resultado tão bom ao acaso. Cada *bit* adicional dobra o tamanho do espaço de busca. Esta pontuação representa um *bit* de probabilidade; um acima de dois elevado a esse montante é a probabilidade de se encontrar um segmento tão ao acaso.

Os *Bit-Scores* representam um nível de probabilidade para comparações entre sequências que é independente do tamanho da pesquisa. O tamanho do espaço de busca é proporcional ao produto do comprimento da sequência de consulta vezes a soma dos comprimentos das sequências no banco de dados.

Este produto, chamado de N nas publicações de Altschul, é multiplicado por um coeficiente K para se obter o tamanho do espaço de busca. Ao se consultar bases de dados de proteínas com consultas de proteínas, o valor de K é de cerca de 0,13. O Netblast usa estimativas de K produzidas antes de executar a consulta, por simulação aleatória (ALTSCHUL et al., 1997).

Há uma probabilidade associada a cada comparação par-a-par na lista e com o alinhamento de cada par de segmento. O número mostrado na lista, conhecido como valor E é a probabilidade de que iria-se observar uma pontuação ou grupo de pontuações tão altas quanto o escore observado puramente por acaso, quando seja feita uma busca em um banco deste tamanho.

Uma busca ideal deveria encontrar acertos que vão do extremamente improvável à aqueles cuja melhor pontuação deve ter ocorrido ao acaso (isto é, com probabilidades aproximando-se de 1.0). Se forem especificados alinhamentos sem a inclusão de lacunas (*ungapped*) para o Netblast, uma terceira coluna de dados será exibida em sua saída sob o título N.

O número na coluna indica quantos HSPs foram envolvidos no cálculo das estatísticas para a sequência. Se o número for maior que 1, os escores de HSPs múltiplos foram combinados para produzir o resultado. No final do arquivo de saída é exibida uma lista de definições de parâmetros, juntamente com algumas informações de rastreamento sobre a pesquisa.

4 VNBlast

Com o amplo desenvolvimento de pesquisas na área de biotecnologia e bioinformática, muitos esforços foram e continuam sendo concentrados na ampliação do conhecimento em pesquisas biomoleculares, pois esta área representa um campo fértil e amplo para o desenvolvimento de novas aplicações (BOONE; UPTON, 2000).

Com o passar do tempo, vários aplicativos foram desenvolvidos por pesquisadores e programadores de sistemas com várias finalidades específicas dentro da bioinformática. Dentre eles é possível destacar o grande projeto ARB (LUDWIG et al., 2004).

Dentre os vários segmentos de pesquisa possíveis nesta área, um procedimento em especial tem atraído a atenção dos pesquisadores: a comparação entre sequências genéticas através do processo de alinhamento. Este procedimento, no aspecto computacional, demandou e continua demandando o desenvolvimento de algoritmos que possam encontrar o melhor alinhamento entre duas ou mais sequências em um tempo adequado.

Uma vez que algoritmos computacionais são usados neste procedimento, surge então a necessidade da análise do custo computacional que a execução destes algoritmos demanda. A técnica de alinhamento local apresentada por Altschul (ALTSCHUL et al., 1990) e previamente descrita neste trabalho mostra que o algoritmo BLAST se consolidou como um dos mais importantes algoritmos para alinhamento local entre sequências da atualidade.

Desde o surgimento do BLAST até hoje, várias implementações deste algoritmo foram feitas através de ferramentas divididas em várias categorias, que vão desde aplicações em modo texto, até a softwares para a internet como a própria página de internet do BLAST oferecido pelo NCBI e o pacote de aplicações em modo console do BLAST (Blastp, Blastx, Blastn, TBlastx, TBlastn, Netblast...) (ALTSCHUL et al., 1997).

Porém, assim como foi descrito no capítulo introdutório deste trabalho, o principal foco dos desenvolvedores externos à NCBI foi o uso do BLAST em modo binário para a geração de alinhamentos através de sua configuração básica e padrão de parâmetros, para em seguida, baseados nos resultados do BLAST, gerar algum tipo de processamento. Boa

parte das ferramentas existentes fazem uso de conjuntos de dados obtidos através de bancos de dados genéticos como o GenBank, sendo que estes conjuntos de dados são baixados de forma manual para o computador que irá efetuar o alinhamento.

É fato que o alinhamento executado sobre bases de dados locais apresenta um maior desempenho quando em comparação a alinhamentos efetuados através da internet. Porém o tamanho atual destes conjuntos de dados (que possuem algumas centenas de gigabytes) e o fato de que estas informações possuem prazo de validade (em função do número diário de submissões de novas sequências), tornam este procedimento pouco interessante para pesquisadores que fazem pequenas, mas frequentes buscas diárias.

A página de internet do NCBI oferece a possibilidade de buscas e alinhamentos através do BLAST, livres de instalação local. Mas devido ao número de acessos diários simultâneos, também são impostas limitações no número de parâmetros e na quantidade de sequências a serem alinhadas.

Como uma interessante alternativa, o NCBI oferece um conjunto de aplicativos que fazem o processo de pesquisa e alinhamento no GenBank de forma remota sem as atuais limitações que a página de internet do NCBI possui. Porém esta suíte de aplicativos disponibilizados pelo NCBI, apesar do grande número de parâmetros disponíveis e dos resultados apresentados, não possui interface gráfica com o usuário, o que torna pequena a interação humano-máquina.

No caso destas aplicações console e em especial o Netblast, existem aproximadamente 40 parâmetros que podem ser definidos através de linha de comando para a potencialização dos resultados, mas que também podem induzir a erros na digitação do comando. Isto certamente acarretará a geração de resultados incorretos ou erros na execução da consulta.

Objetivando oferecer o melhor destas duas realidades, ou seja, oferecer uma interface gráfica para a internet com grande interação com o usuário, além de oferecer os parâmetros disponíveis na aplicação em linha de comando, o VNBlast surge como uma interessante alternativa para pesquisadores e estudantes, possibilitando além de outras, a possibilidade de buscas em lote.

O VNBlast foi desenvolvido sob um conceito de aplicações para a internet chamado R.I.A. - (*Rich Internet Application*). As aplicações desenvolvidas sob este conceito apresentam as características que aplicações *desktop* possuem, como os tradicionais formulários e uma ampla gama de componentes gráficos de formulários, como caixas de combinação

inteligentes, grades, caixas de texto, botões entre outros, além da possibilidade de execução de multimídia em *streamings*, tudo sendo executado através de um navegador de internet.

4.1 Ferramentas Utilizadas no Desenvolvimento do VNBlast

A escolha das ferramentas de programação para o desenvolvimento do VNBlast foi um passo muito importante neste projeto. A linguagem escolhida deveria ser capaz de lidar com situações distintas: os ambientes web distribuídos e a capacidade de gerenciar aplicações do tipo console, tais como o Netblast.

Para este fim, o Java EJB (ORACLE, 2011) foi escolhido para ser a linguagem de programação principal e o JBoss (COMUNITY, 2011) como o servidor de aplicação, também conhecido como *container* EJB. A utilização do JBoss foi necessária uma vez que o VNBlast foi desenvolvido como uma aplicação em camadas.

No caso do VNBlast, as camadas são as seguintes: a camada *service*, a camada *view* e a camada *remote*. A camada *service*, representada pelo JAVA EJB é a camada responsável pelos processamentos mais pesados dentro da aplicação tais como interfacear o aplicativo NETBLAST, receber as solicitações enviadas pelo JBoss e processá-las, assim como receber o retorno da consulta e enviar os resultados ao cliente.

A camada *view*, representada pelo Adobe FLEX (ADOBE, 2011) é a camada responsável pela interface gráfica com o usuário. Embora exista pouco processamento nesta camada, este se restringe apenas a validações de interface gráfica e tratamento de parâmetros em componentes do Flex.

Por fim, a camada *remote* é a camada que efetua o enlace entre as camadas *view* e *service*, ou seja, é a comunicação entre o JAVA EJB e o Adobe FLEX. O Flex é uma linguagem criada para o desenvolvimento de interfaces gráficas para a internet, compatível com o JAVA. O Flex apresenta bons recursos para o projeto de interfaces amigáveis sob o conceito RIA - *Rich Internet Applications*. Recursos como formulários, caixas de combinação, *tooltips* e grades estão disponíveis nesta ferramenta. O VNBlast foi desenvolvido como um projeto de código aberto e partindo deste pressuposto, as ferramentas escolhidas e utilizadas são livres para uso na internet.

Um dos objetivos do projeto VNBlast foi a reutilização de uma aplicação estável pré-existente na sua forma binária, reduzindo deste modo, significativamente o tempo

de desenvolvimento da ferramenta. Mas para isso, um ponto importante a ser verificado foi a capacidade de gestão de aplicações externas pela linguagem de programação escolhida.

O controle do exato momento da inicialização e finalização do aplicativo de console é de fundamental importância neste caso, uma vez que é o sistema operacional quem gerencia o status dos processos que são executados no equipamento. O Java possui a classe java.lang.Runtime, que troca informações com o sistema operacional, permitindo a visão do status dos processos inicializados à partir da aplicação. As subsessões a seguir apresentarão as ferramentas utilizadas para o desenvolvimento do VNBlast em maiores detalhes.

4.1.1 JAVA EJB

A tecnologia Enterprise JavaBeans (EJB) (ORACLE, 2011) é a arquitetura de componentes do lado do servidor para a Plataforma Java na versão Enterprise Edition (Java EE). A tecnologia EJB permite o desenvolvimento rápido e simplificado de aplicativos distribuídos, transacionais, seguros e portáteis baseados na tecnologia Java.

A especificação EJB 3.0 *Final Release* define o novo quadro simplificado de APIs EJB especialmente criadas para uma maior facilidade no desenvolvimento de programas. Nesta especificação também está incluída a nova API Java Persistence para a gestão de persistência e mapeamento de objetos/relacionamentos com o Java EE e o Java SE.

A API - *Java Persistence* é a API padrão para o gerenciamento de persistências e mapeamento de objetos e relacionamentos. Esta API fornece um objeto de mapeamento relacional para desenvolvedores de aplicações usando um modelo de domínio Java para gerenciar um banco de dados relacional. A *Java Persistence API* é parte da plataforma Java EE, mas que também pode ser utilizada em ambientes Java SE.

A especificação EJB 2.1, criada sob a *Java Community Process (JCP)*, aumenta a arquitetura EJB com suporte para serviços Web, tornando-o mais fácil de implementar e implantar aplicativos de serviços Web baseados em tecnologia Java.

EJB ou Enterprise JavaBeans é um dos principais componentes da plataforma J2EE (Java 2 Enterprise Edition). É um componente do tipo servidor que é executado no *container* do servidor de aplicação. Atualmente ele encontra-se na versão 3.1 e o seu futuro é definido conjuntamente entre grandes empresas como IBM, Oracle e HP, além de uma vasta comunidade de programadores numa rede mundial de colaboração sob o portal do JCP. A grande mudança entre a versão 2.1 e a versão 3.0 é a introdução de anotações Java. As anotações facilitam o desenvolvimento, diminuindo a quantidade de código e o uso de arquivos

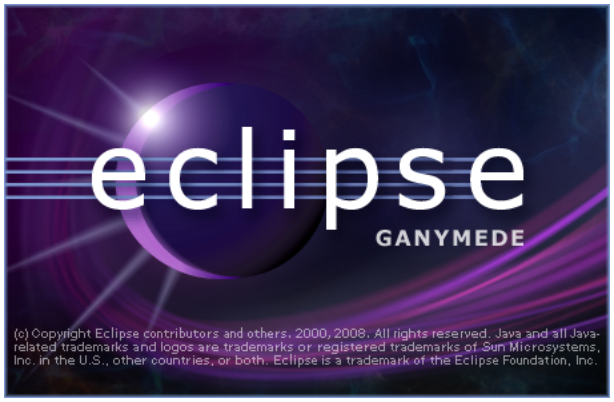


Figura 4.1: Logomarca oficial do Eclipse na versão Ganymede, utilizado no desenvolvimento do VNBlast.

de configuração XML.

A plataforma J2EE provê algumas facilidades dedicadas à camada de regra de negócio e para o acesso a banco de dados. Através do EJB o programador utiliza a infraestrutura do servidor de aplicação focada no desenvolvimento de aplicações de missão crítica (de alta importância para a empresa) e de aplicações empresariais em geral.

Um importante aprimoramento na tecnologia EJB é a adição da nova API *Java Persistence*, o que simplifica o modelo de persistência da entidade e acrescenta recursos que não estavam contidos na tecnologia EJB 2.1.

Além de simplificar o modelo de persistência da entidade, a *Java Persistence API* padroniza o mapeamento objeto-relacionamento. Em suma, o EJB 3.0 possibilita o desenvolvimento mais rápido de aplicações.

Com a inclusão da *Java Persistence API*, o EJB 3.0 também oferece aos desenvolvedores um modelo de programação de entidades que é tanto mais fácil de usar quanto mais poderoso. A *Java Persistence API* baseia-se em ideias de frameworks de persistência de liderança e APIs como *Hibernate, TopLink* da Oracle e o *Java Data Objects* (JDO), assim como na persistência EJB anterior gerenciada por contêiner.

A Figura 4.1 apresenta a Logomarca oficial do Eclipse na versão Ganymede, versão esta que foi utilizada para o desenvolvimento JAVA do VNBlast. A Figura 4.2 apresenta a IDE *open-source* Eclipse de desenvolvimento Java Enterprise Edition (Java EE), utilizada para desenvolver o código Java do VNBlast.

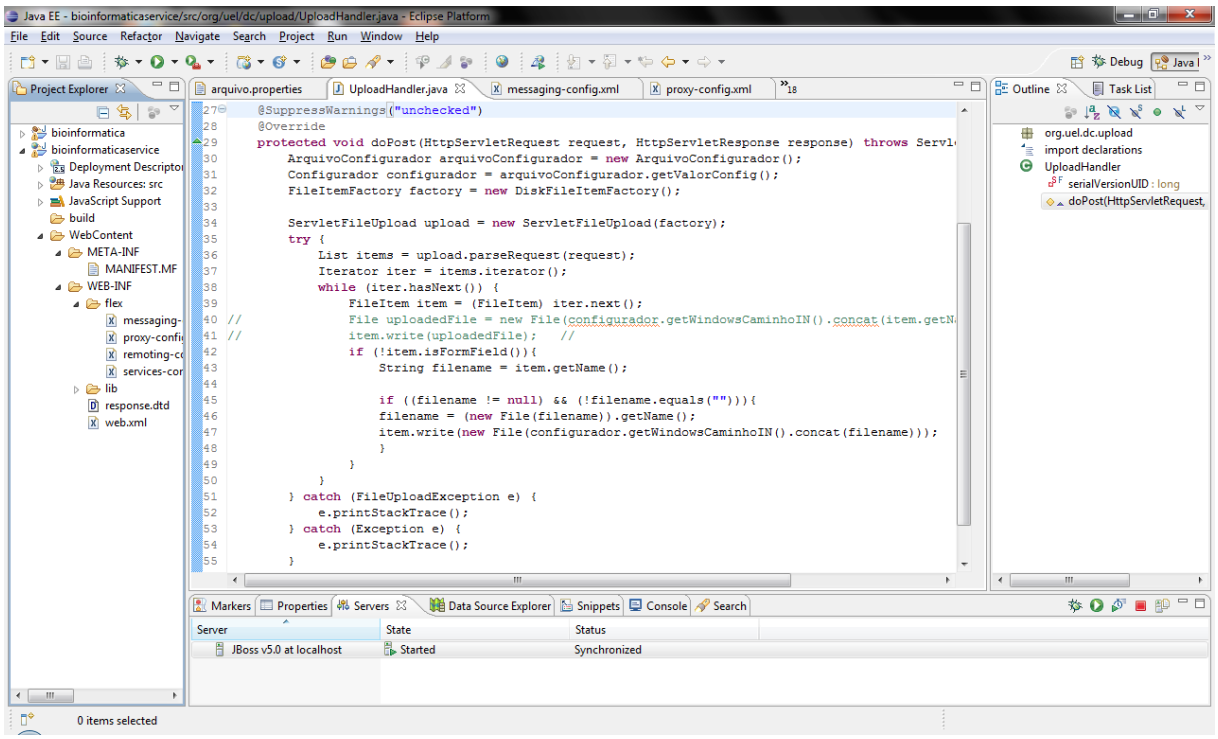


Figura 4.2: IDE de desenvolvimento Eclipse.

4.1.2 JBOSS

O JBoss (COMUNITY, 2011) é um dos servidores de aplicação JEE de maior sucesso atualmente. Sob a licença de projeto de código aberto, o JBoss compete de frente com gigantes como o IBM Websphere (IBM, 2011).

O JBoss existe desde 1999, inicialmente conhecido como EJBoss ou EJB Open Source Server, posteriormente foi renomeado para JBoss. Ganhou fama juntamente com o Tomcat (TEAM, 2011a) na briga existente entre os servidores Java atuais. O resultado de sua popularidade foi o anuncio de que em 2006 a Red Hat assinou um acordo definitivo para adquirir a fornecedora de software de código aberto JBoss por cerca de 350 milhões de dólares em dinheiro e ações, publicado a algum tempo atrás pela revista eletrônica IDG Now. Isto depois de alguns meses de especulações de que a gigante Oracle estaria negociando a aquisição da JBoss Group.

Distribuído sob a licença GNU, com status de 100% gratuito, o JBoss é implementado totalmente em Java. O custo zero não é o principal diferencial desse servidor Java. Há outros atrativos como o *hot deploy*, que é um conceito que todos os servidores J2EE utilizam para tratar a distribuição e recarga de classes Java em tempo de execução permitindo que os administradores alterem versões de seus objetos sem a necessidade de reiniciar o servidor de aplicações e *proxies* dinâmicos, onde todas as invocações aos métodos do objeto real são

Figura 4.3: Seleção de idiomas do instalador do JBoss, incluindo o português do Brasil.

interceptadas e o proxy decide se realmente delega ao objeto real à invocação do método, ou não (ORACLE, 2011). É importante notar o fato do JBoss ser relativamente leve, com baixa requisição de memória e espaço em disco, sem comprometer sua performance.

O JBoss Application Server (JBoss AS) (ORACLE, 2011) é certamente o mais conhecido servidor de aplicações Java EE livre, competindo de igual para igual no mercado com produtos consagrados como o Weblogic da BEA. A liderança tecnológica do JBoss AS é confirmada pela participação ativa dos seus desenvolvedores nas definições da versão 5 da plataforma Java EE, em especial na especificação EJB 3 (ORACLE, 2011).

Com a compra da JBoss Group pela Red Hat e a parceria com a Exadel, a JBoss AS passou a ser a base de uma família de produtos que cobre todas as demandas de desenvolvimento e infra-estrutura de produção para aplicações Java EE. Isso sem abrir mão do modelo de negócios *Open Source Profissional* - isto é, os produtos com a marca JBoss continuam sendo fornecidos integralmente sob licenças livres, e a receita provêm integralmente da prestação de serviços sobre os produtos. Todo o poder, flexibilidade e confiabilidade do JBoss AS vêm em um pacote muito simples de instalar. Rodar os primeiros *servlets*, EJBs e consumidores de mensagens JMS é igualmente simples.

Mas, embora haja facilidade em se passar pelos passos iniciais com a JBoss AS, não se pode subestimar o esforço e conhecimentos necessários para se manter um ambiente de produção para aplicações Java EE. A Figura 4.3 apresenta a opção no instalador de seleção do idioma de instalação desejado para o JBoss. Esta versão do pacote de instalação apresenta a vantagem de incluir o idioma português-BR nativamente.

A Figura 4.4 apresenta algumas opções de instalação disponíveis para o JBoss que variam de opções mais básicas e compactas de instalação até opções de instalação mais completas.

A Figura 4.5 exibe algumas opções de pacotes disponíveis a serem incluídos na instalação do JBoss. Dependendo do tipo de uso do servidor, será necessária a adição de

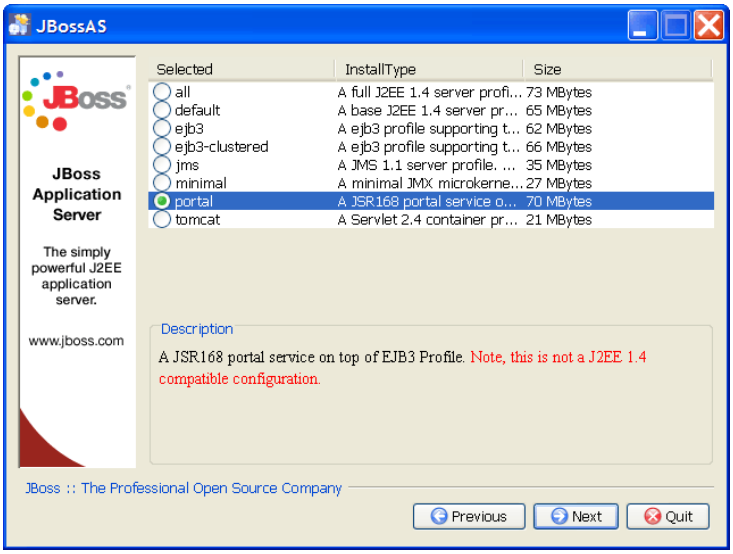


Figura 4.4: Seleção das opções de instalação do JBoss.

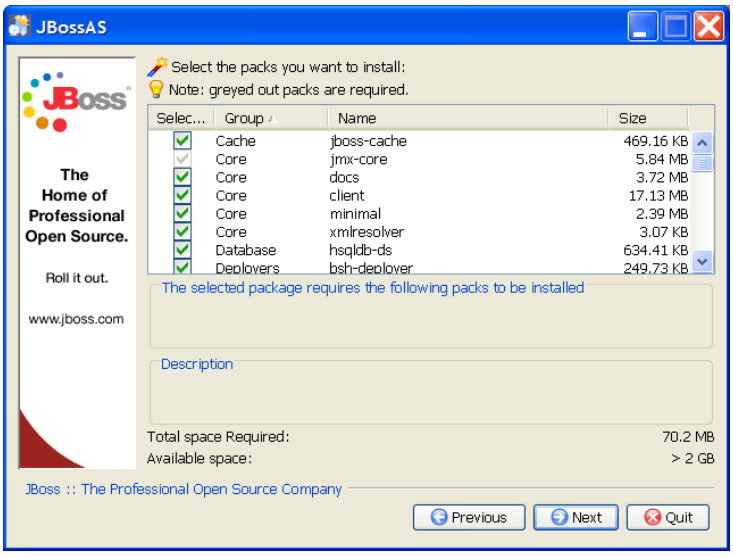


Figura 4.5: Seleção dos pacotes do JBoss a serem instalados.

alguns pacotes opcionais.

A Figura 4.6 exibe as opções de configuração de segurança do JBoss, aplicados através do JMX.

A Figura 4.7 apresenta os parâmetros de configuração necessários para o funcionamento correto do JBoss, que serão incluídos no conjunto de variáveis de ambiente do sistema operacional de forma automática pelo instalador.

Por fim, a Figura 4.8 apresenta o portal de configurações do JBoss, através do qual é possível alterar alguns parâmetros de configuração do JBoss após a sua instalação no servidor de destino.

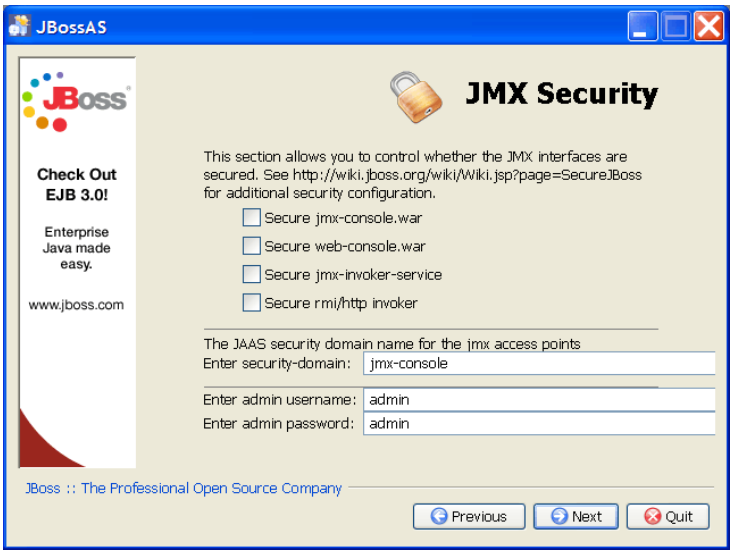


Figura 4.6: Seleção das opções de segurança do JMX.

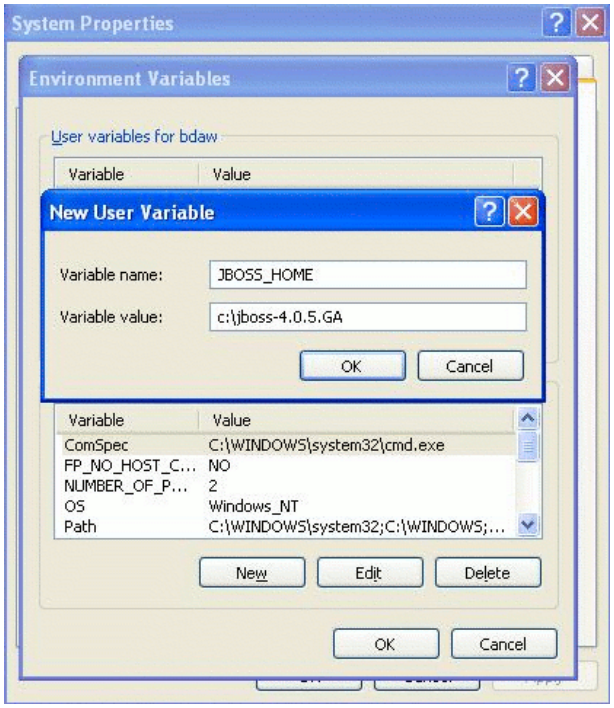


Figura 4.7: Configuração das variáveis de ambiente do sistema operacional, incluindo os parâmetros necessários do JBoss.

Figura 4.8: Portal de configurações do JBoss, acessado através de um navegador de internet.

4.1.3 ADOBE FLEX

Assim como o Adobe Flash, o Adobe Flex cria arquivos SWF que são processados pelo Flash Player. No entanto, o Flex é principalmente uma ferramenta criada para desenvolvedores, e a forma com que as interfaces RIA - *Rich Internet Application* são criadas no Flash é bem diferente da forma com que as interfaces RIA são criadas através do Flex.

Todo o desenvolvimento do Flex é baseado em uma estrutura que fornece componentes de interface reutilizáveis e extensíveis, serviços de recuperação de dados, funcionalidades de manipulação de eventos e muito mais.

Com o Flex é possível criar RIAs num ambiente centrado em código familiar (um pouco parecido com o Java). Com o Adobe Flex ainda é possível utilizar recursos de aplicações Flash, que incluem a habilidade para projetar e implementar interfaces de grande interação com o usuário sem a preocupação com possíveis limitações do navegador de internet. Esta ferramenta possibilita um ambiente de execução com alcance para quase a totalidade dos usuários da Internet.

O Flex possui uma poderosa linguagem de programação chamada ActionScript que tem a capacidade de integrar mídias de alto padrão, como *streamings* de vídeo e som. No Adobe Flash, o ambiente de desenvolvimento gira em torno de uma metáfora de linha do tempo e ferramentas de desenvolvimento visual.

Devido a esta ênfase, e apesar da evolução da linguagem de programação Flash com a inclusão do ActionScript, uma linguagem totalmente orientada a objetos, ECMA-4 compatível, o Flash tem sido tradicionalmente visto como uma ferramenta de desenvolvimento para a criação de animações.

Ao longo dos anos, muitos desenvolvedores perceberam que poderiam usar o Flash para criar RIAs poderosas, mas outros têm encontrado dificuldades na compreensão do uso da IDE do Flash para este fim específico (ADOBE, 2011).

Com o Adobe Flex, o desenvolvimento de RIAs se tornou mais simples e poderoso através de uma IDE dedicada a esta finalidade. O Flash e Flex podem trabalhar em conjunto, possibilitando a criação de páginas da web ainda mais ricas. Na realidade, o Flex pode trabalhar diretamente com outros produtos da Adobe Create Suite. Designers e programadores podem facilmente criar seus próprios ambientes e então integrar seu trabalho para estabelecer uma série inteiramente nova de RIAs.

Flex é voltada para programadores, ao invés de designers. A linguagem de

programação principal Flex, ActionScript 3, é orientada a objetos, por isso é uma linguagem recomendada para desenvolvedores que já possuam alguma experiência com os conceitos de O.O. Após a decisão por parte do desenvolvedor do uso do SDK do Flex gratuito, ou do Flex Builder IDE (proprietário), já será possível o início do desenvolvimento de uma interface RIA com o Flex.

Se a escolha foi feita pelo SDK gratuito, será necessária a utilização de um compilador independente para a compilação do código nativo em um arquivo SWF, uma vez que na versão gratuita o compilador não está incluído. Já o Flex Builder pode ser configurado para compilar o código automaticamente e também criar a detecção do navegador de código necessário para aplicações do lado do cliente.

O Flex faz uso de duas linguagens: O MXML (ADOBE, 2011), uma linguagem de marcação baseada em XML, que é usado principalmente para elementos de *layout* de exibição do aplicativo e o ActionScript (ADOBE, 2011), que é uma linguagem de *script* compatível com o padrão ECMA (ADOBE, 2011), além de ser uma linguagem de programação orientada a objeto que é usada principalmente para a lógica da aplicação.

Durante a compilação, o código MXML é traduzido em código ActionScript e, em seguida, todo o código ActionScript é compilado em arquivos binários SWF. O arquivo SWF pode ser carregado para o servidor web, onde é então servido com base na solicitação do usuário.

Para o desenvolvimento deste trabalho, a Adobe gentilmente forneceu uma licença de uso da versão 3.0 *Educacional* da IDE de desenvolvimento do Flex, sendo a interface gráfica do VNBlast totalmente desenvolvida à partir desta.

A Figura 4.9 apresenta a sequência entre o desenvolvimento de uma aplicação através do Flex e a visualização da mesma pelo usuário cliente.

A Figura 4.10 exibe a logomarca de abertura do Adobe Flex, concedido pela Adobe para uso neste projeto.

A Figura 4.11 apresenta a IDE de desenvolvimento do Flex (que possui aparência semelhante à IDE Eclipse) no modo de exibição de código fonte.

A Figura 4.12 apresenta a IDE de desenvolvimento do Flex no modo de exibição de formulários, apresentando o formulário de consultas do VNBlast.

A Figura 4.13 apresenta a IDE de desenvolvimento do Flex no modo de exibição de formulários, apresentando o formulário inicial do VNBlast.

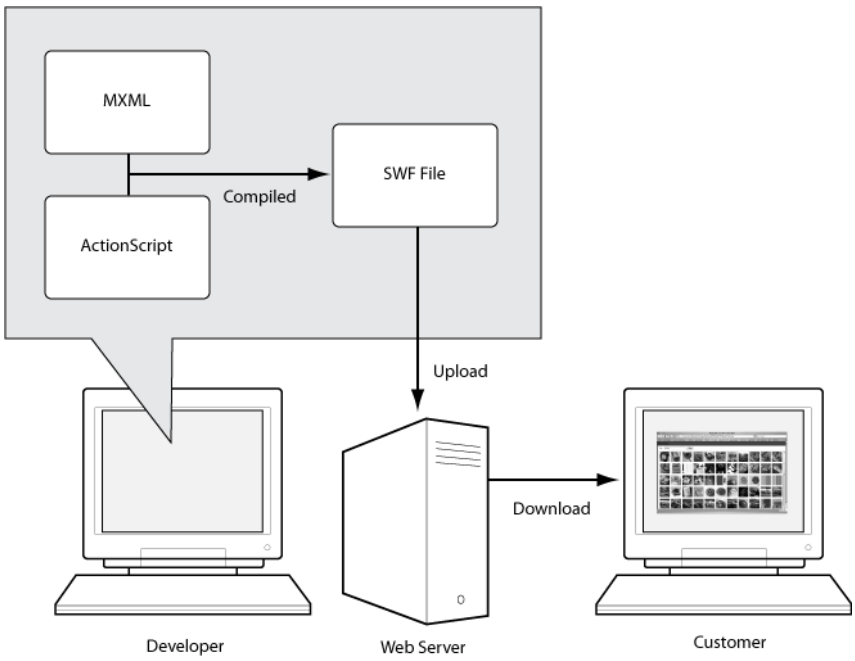


Figura 4.9: Etapas entre o desenvolvimento da interface gráfica e a visualização final pelo usuário. Imagem obtida do site oficial da Adobe (ADOBE, 2011)

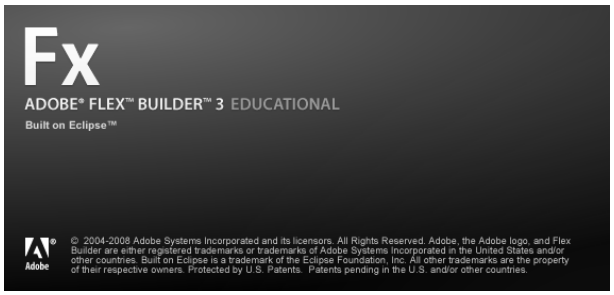


Figura 4.10: Logomarca de abertura do Adobe Flex

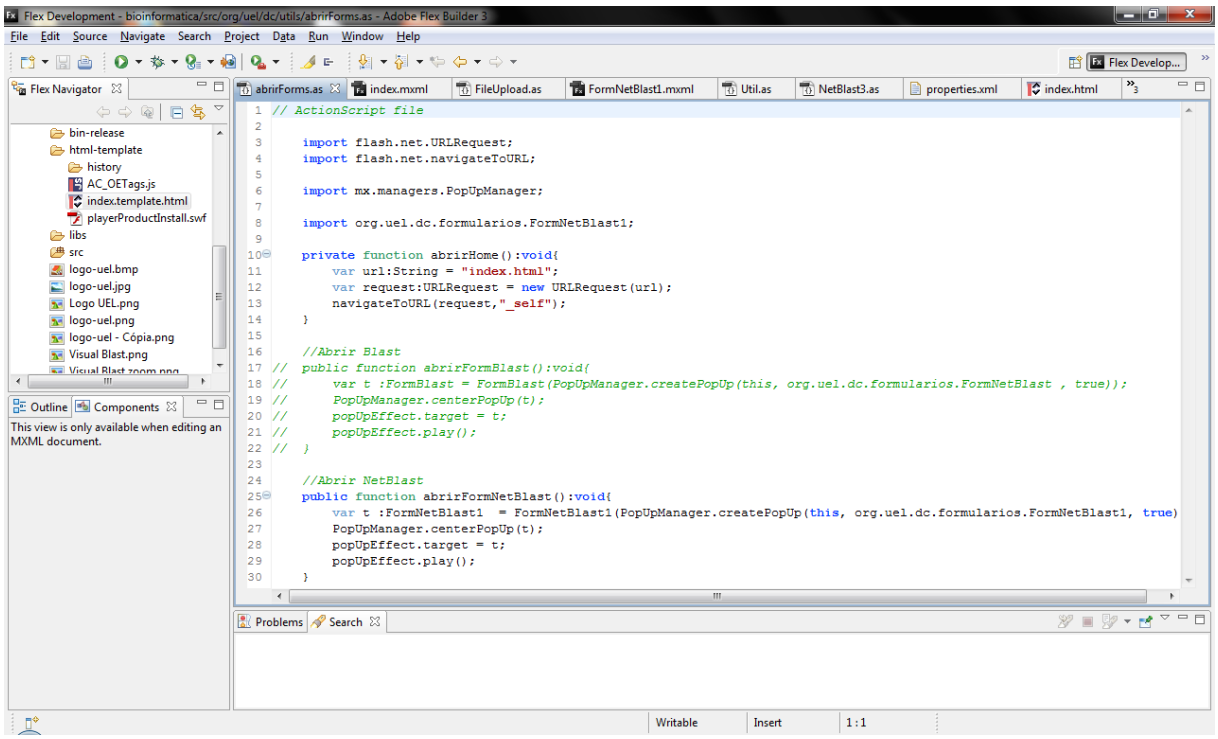


Figura 4.11: IDE de desenvolvimento do Flex (versão educacional) no modo de edição de código fonte.

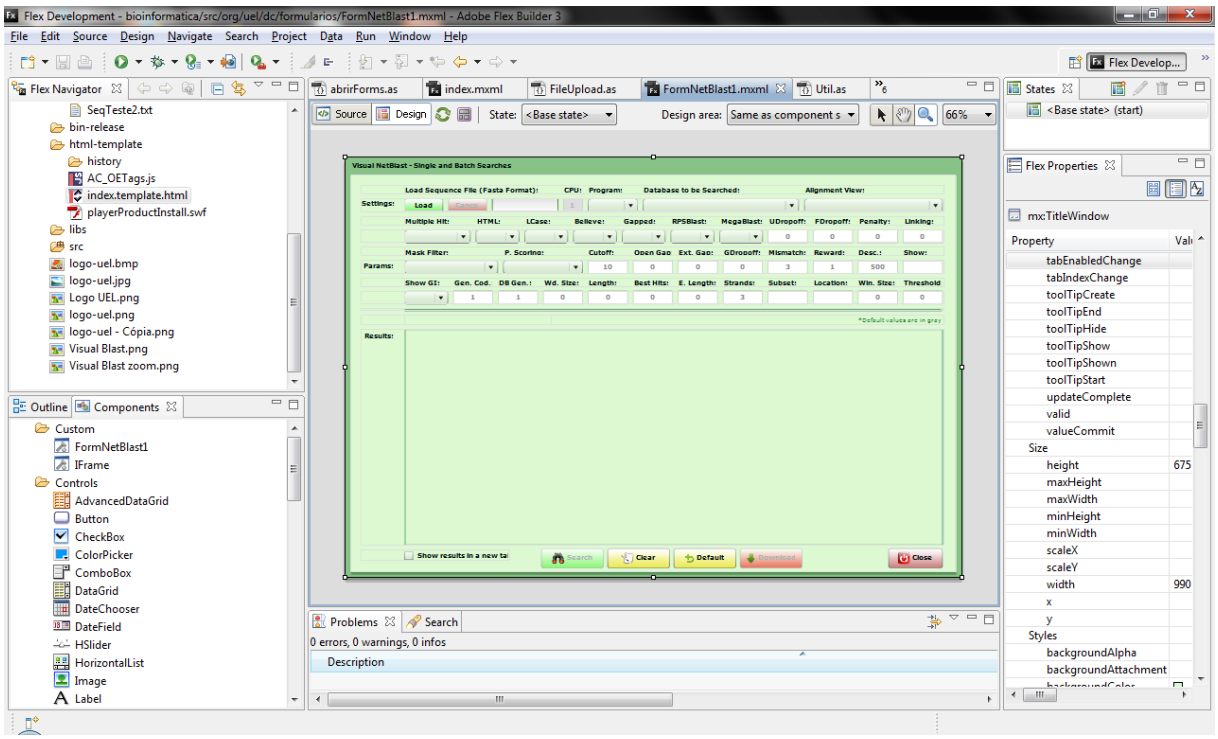


Figura 4.12: IDE de desenvolvimento do Flex (versão educacional) no modo de exibição de formulários, apresentando o formulário de consultas do VNBlast.

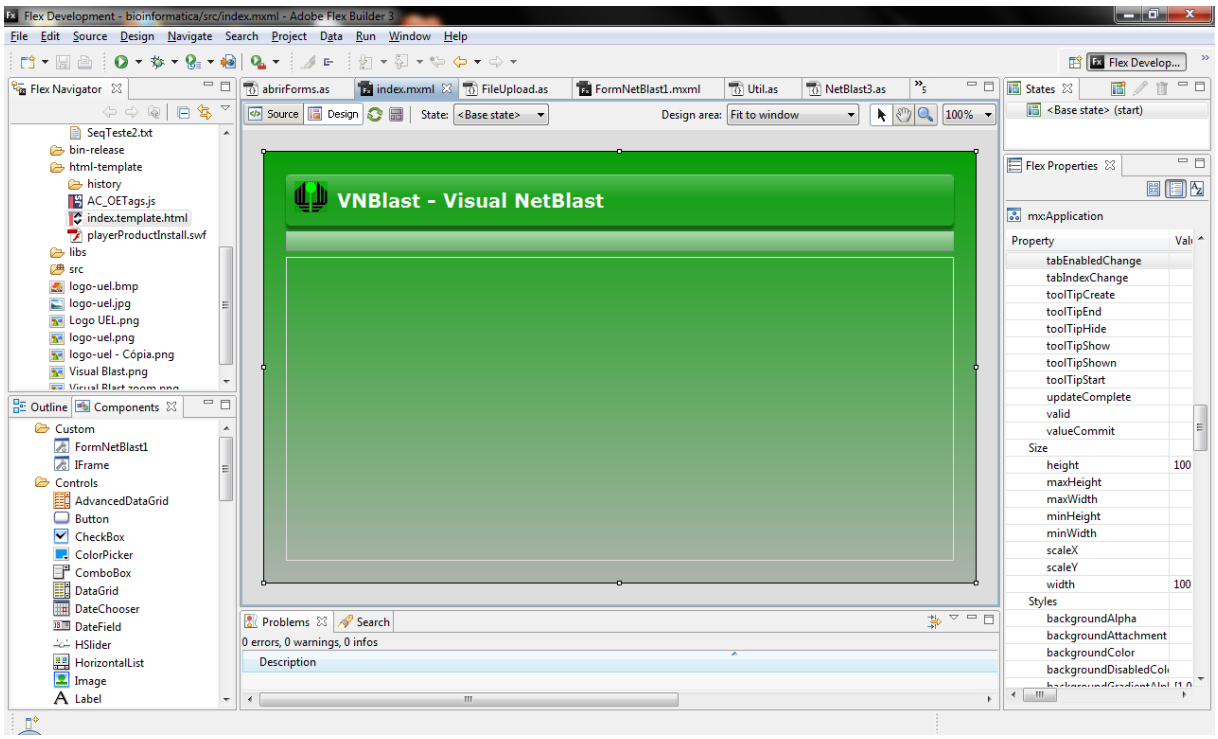


Figura 4.13: IDE de desenvolvimento do Flex no modo de exibição de formulários, apresentando o formulário inicial do VNBlast.

4.2 Integração entre as Ferramentas

As ferramentas selecionadas para o desenvolvimento do VNBlast, foram desenvolvidas originalmente de modo a permitirem uma integração harmoniosa entre si. O Adobe Flex, responsável pela camada *view* (que é a própria interface gráfica que irá interagir com o usuário), é uma ferramenta que trabalha com o *ActionScript* e o MXML, sendo que o *ActionScript* é uma linguagem de programação desenvolvida para processamentos relacionados à própria interface gráfica do Flex, não sendo possível, por exemplo, a persistência de dados em algum banco de dados através do ActionScript.

Porém o Flex permite uma integração de chamadas entre a aplicação desenvolvida em Flex e classes que tenham sido desenvolvidas em outras linguagens de programação. No caso do VNBlast, a linguagem de programação através da qual foram desenvolvidas as classes e os principais métodos do sistema foi o Java EJB.

A linguagem Java dispensa maiores comentários, uma vez que é notavelmente uma das maiores linguagens de programação atualmente em uso em aplicações para internet, tendo ademais a capacidade para desenvolvimento de aplicações também para uso local, como por exemplo, aplicações *desktop*.

O JBoss, que é um bem conceituado servidor de aplicações, foi escolhido

para gerenciar o VNBlast. O primeiro motivo é o tempo de existência e maturidade deste servidor, além do fato de que este foi desenvolvido totalmente em Java (e também para Java), garantindo uma melhor estabilidade na aplicação.

Todas as ferramentas escolhidas são caracterizadas como de código aberto, com exceção da IDE do Flex que é uma ferramenta proprietária da Adobe, mas no desenvolvimento do VNBlast, esta licença foi concedida de forma gratuita pela Adobe.

O VNBlast foi escrito como uma aplicação do tipo *server-side*, ou seja, praticamente todo o processamento é executado no lado servidor. Neste caso, mais especificamente, o servidor do VNBlast também se transforma em um cliente em um dado momento.

Em uma primeira etapa o usuário acessa o servidor através de um navegador da web, monta a consulta informando os parâmetros e selecionando o arquivo com a sequência de critério no lado cliente; por fim submete a consulta ao servidor.

O servidor recebe a requisição de consulta, os parâmetros selecionados da consulta e o arquivo contendo a sequência submetidos pelo usuário, monta a consulta no formato correto com todos os parâmetros necessários e submete a consulta aos servidores do GenBank através de serviços da web (*webservices*).

O servidor do GenBank, por sua vez, ao receber a solicitação de busca e alinhamento, redireciona a busca para o(s) bancos de dados adequados em seu *data center*. A localização das sequências mais similares ocorre através da comparação por alinhamento usando o algoritmo BLAST. Após ser encerrada a busca, o sistema de gerenciamento do GenBank monta o resultado, exibindo o alinhamento e a tabela com os valores estatísticos de resultado da consulta para retorno.

Na próxima etapa, o GenBank envia o pacote com os resultados da busca ao servidor do VNBlast através do *webservice* solicitado, que por sua vez recebe o pacote de retorno e monta os resultados da consulta em um arquivo.

O arquivo, gerado no servidor do VNBlast é então submetido para o lado cliente da aplicação, sendo então visto pelo usuário através do formulário de consultas do VNBlast ou através de uma nova aba do navegador, caso esta opção tenha sido selecionada pelo usuário. A Figura 4.14 apresenta o diagrama de fluxo de informações entre o usuário solicitante, o VNBlast e o GenBank.

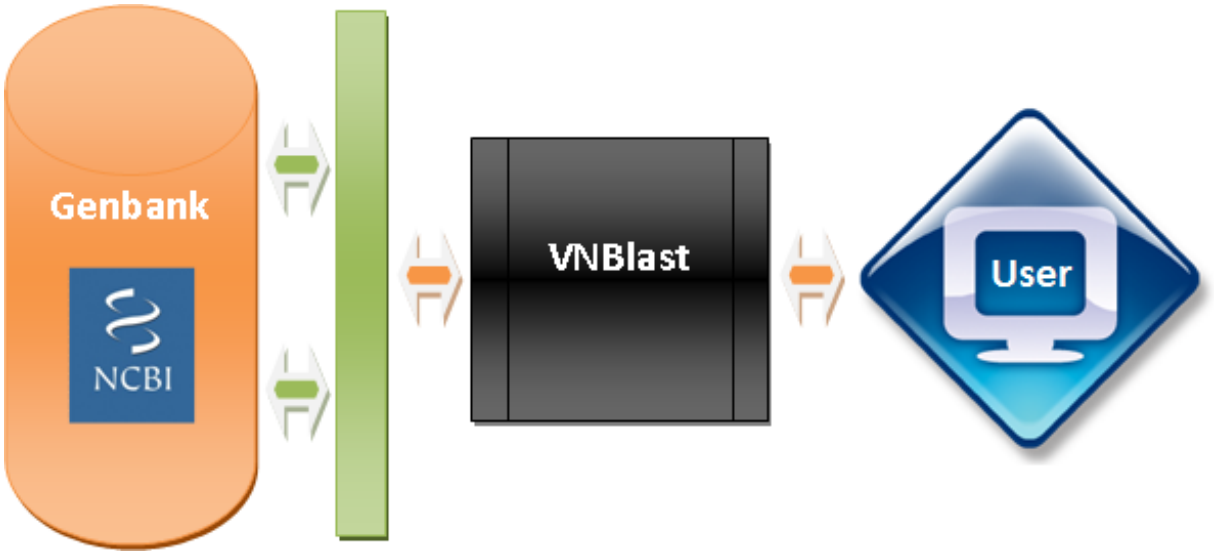


Figura 4.14: Diagrama de fluxo de informações entre o usuário, o VNBlast e o GenBank.

5 Resultados Experimentais

A ampliação gradual da velocidade das conexões de Internet disponíveis, aliada à redução também gradual do custo de acesso a estas conexões favoreceu o surgimento de novas tecnologias e a quebra de paradigmas no mecanismo de funcionamento da internet (PORCINO; HIRT, 2003). *Streamings* de vídeo de alta resolução, *downloads* a altas taxas de transferência, voz e imagens em alta definição e em tempo real são tecnologias já disponíveis (PORCINO; HIRT, 2003).

Com essa evolução tecnológica no meio físico da internet, surgiu um novo conceito de aplicações sendo executadas através da internet: as aplicações ricas para a internet. R.I.A.s, como são conhecidas, são aplicações desenvolvidas para serem executadas em navegadores de internet como páginas da internet, mas que possuem características muito peculiares de aplicações originalmente desenvolvidas para execução local *desktop*, como formulários, botões, grades, caixas de combinação entre outros, além da possibilidade de transparência nos formulários e componentes utilizados.

A utilização dos navegadores de internet para acesso e exibição do sistema possibilita uma maior amplitude de uso para o VNBlast, pois elimina a necessidade de instalação local da aplicação para o posterior uso.

O VNBlast é uma aplicação para a internet que utiliza os navegadores atualmente disponíveis como o mecanismo de exibição de sua interface gráfica, exigindo como pré-requisito apenas que o *plug-in* do Adobe Flash (ADOBE, 2011) esteja instalado. Segundo a própria Adobe em seu site, cerca de 99% dos computadores atualmente conectados na internet possuem o *plug-in* do flash instalados e uma vez que este *plug-in* esteja instalado e atualizado, torna-se então possível o acesso e utilização do VNBlast por parte do cliente.

Como o Adobe Flex possibilitava o desenvolvimento de aplicações R.I.A., e estas características eram extremamente úteis e interessantes para aplicações com as características do VNBlast, então o VNBlast foi criado sob este paradigma.

O VNBlast é constituído de dois formulários. O formulário principal, é exibido

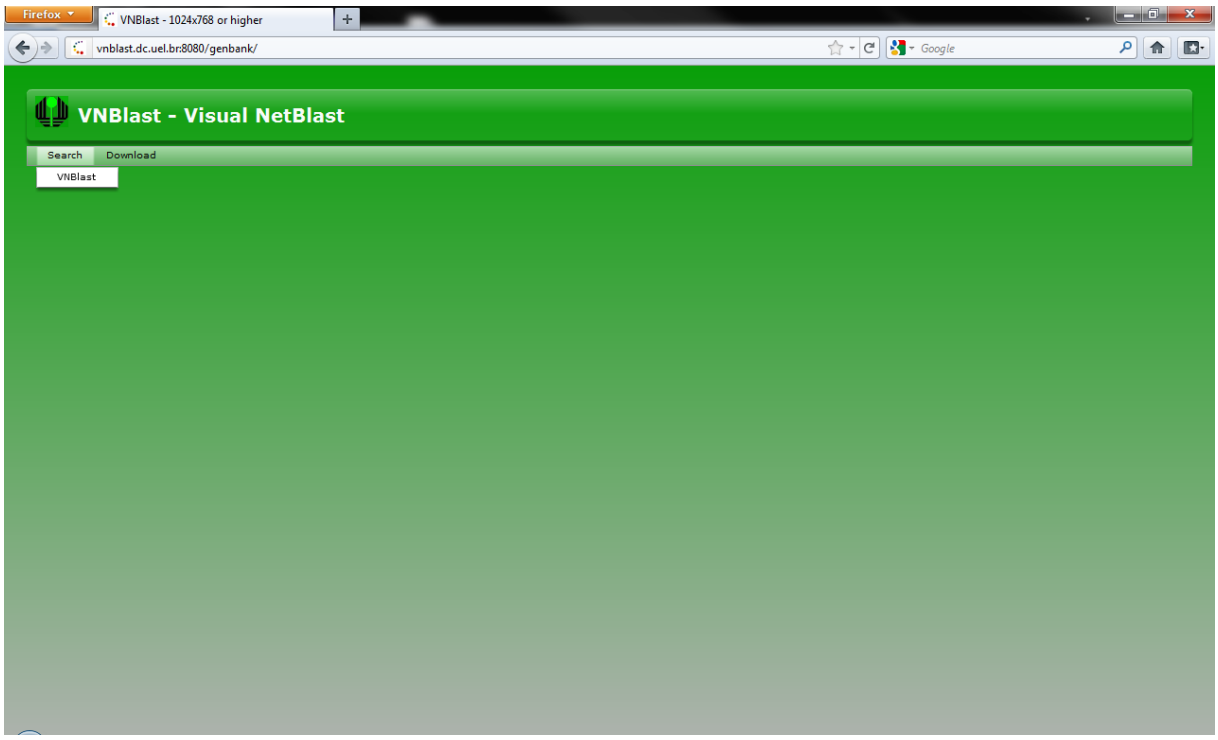


Figura 5.1: Formulário principal do VNBlast, que contém os menus do programa no estilo *pull-down*

de modo maximizado dentro do navegador da internet. Quando o usuário informa o endereço do servidor do VNBlast em seu navegador, então este formulário principal é exibido.

Dentro do formulário principal é possível ter acesso a itens como o formulário de consultas e alinhamentos, efetuar downloads etc. No desenvolvimento deste formulário principal foi feita a opção por um visual mais limpo (menos poluído), sendo exibido apenas o menu no estilo *pull-down* (estilo característico de menus de aplicações do tipo *desktop*), onde em um primeiro momento são exibidos apenas os menus principais e os sub-menus surgem no momento em que o usuário seleciona um dos menus principais. O plano de fundo do formulário principal é exibido na cor verde em degradê.

A Figura 5.1 apresenta o Formulário principal do VNBlast, que contém os menus do programa no estilo *pull-down* apresentando o menu de exibição do formulário de consultas e alinhamentos. A Figura 5.2 também apresenta o Formulário principal do VNBlast, mas agora mostra o menu com a opção de download de um arquivo que contém uma sequência de exemplo para ser inserida no formulário de consulta do VNBlast e apresentar um alinhamento no GenBank.

Quando a opção *Download - Sample Fasta Input File* é selecionada, o VNBlast exibe uma janela solicitando um local para que o arquivo texto contendo a sequência

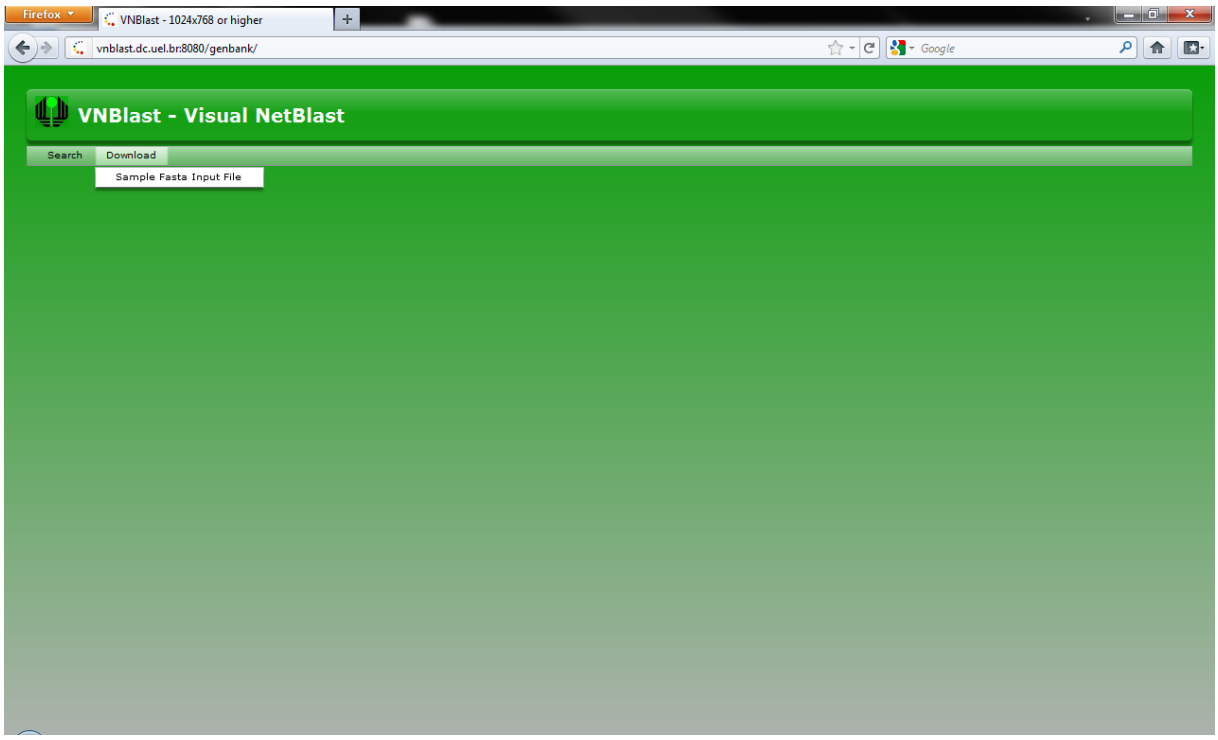


Figura 5.2: Formulário principal do VNBlast, que contém os menus do programa no estilo *pull-down*

de exemplo seja salvo, conforme apresenta a Figura 5.3. O arquivo de exemplo baixado do site do VNBlast contém uma pequena sequência no formato FASTA (PEARSON; LIPMAN, 1988) de um fungo chamado de *Ajellomyces capsulatus var. farciminosus 18s* que poderá ser submetida para a verificação de alinhamentos no GenBank. O arquivo contendo a sequência de bases nitrogenadas deste organismo pode ser visto na Figura 5.4.

Quando o usuário escolhe a opção Search - VNBlast conforme pode ser visto na Figura 5.1, o formulário de busca e alinhamento do VNBlast é apresentado, conforme pode ser visto na Figura 5.5. Este formulário contém desde a opção de seleção do arquivo de critério de busca, até a opção de seleção dos parâmetros de busca e alinhamento desejados.

Os componentes do formulário de busca e alinhamento do VNBlast foram distribuídos de modo a formarem uma sequência lógica de passos a serem executados pelo usuário para a construção de uma consulta no VNBlast. O primeiro passo a ser dado é a seleção do arquivo que contém a sequência de critério no formato FASTA, conforme apresenta a Figura 5.6. O botão *Load* faz com que o VNBlast exiba uma janela solicitando o local onde o arquivo se encontra e o nome do arquivo que contenha a sequência.

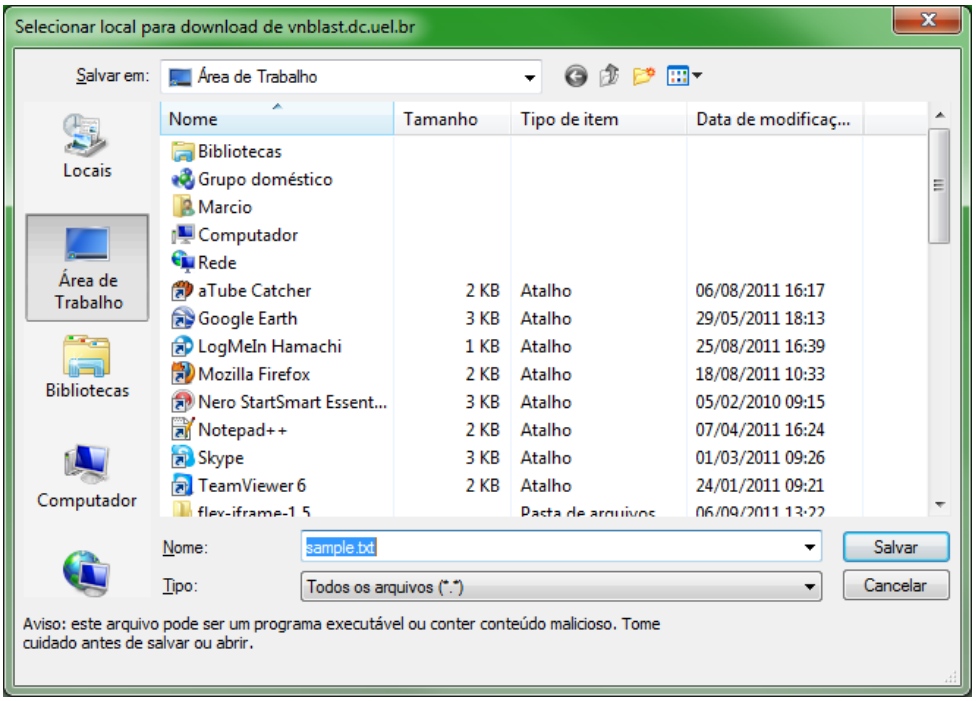


Figura 5.3: Caixa de controle solicitando um local de destino para a gravação do arquivo de exemplo sample.txt

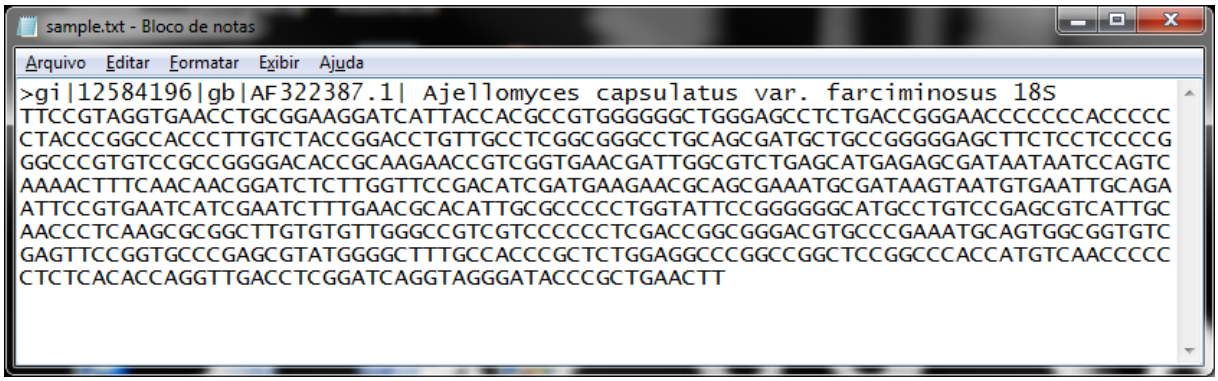


Figura 5.4: Arquivo de exemplo baixado do site do VNBlast, que contém a sequência no formato FASTA do fungo *Ajellomyces capsulatus var. farciminosus 18s*.

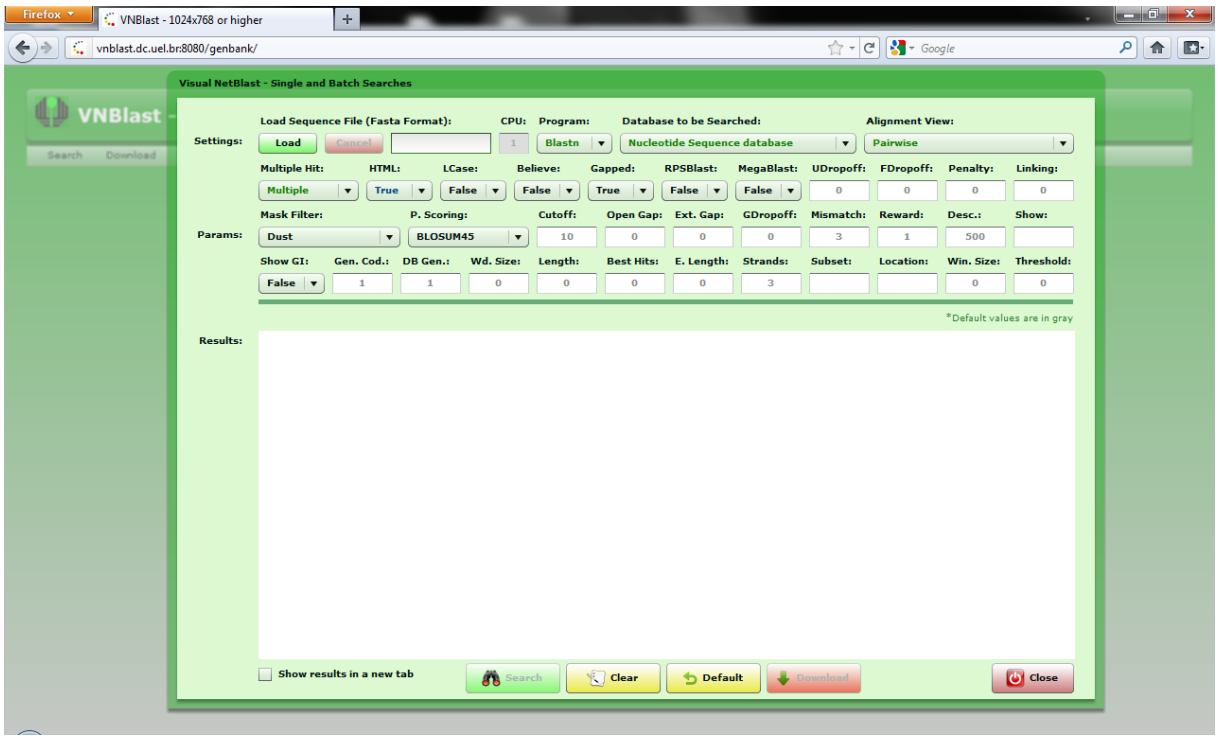


Figura 5.5: Formulário de busca e alinhamento do VNBlast.

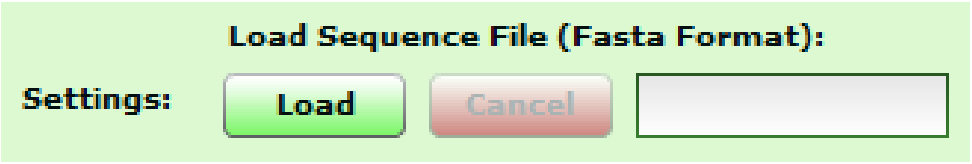


Figura 5.6: Botões para a seleção e cancelamento da seleção do arquivo contendo a sequência FASTA de critério para a consulta.

Caso o arquivo de critério selecionado possua um tamanho grande e o processo de *upload* do arquivo seja muito demorado, então o botão *Cancel* será habilitado, permitindo ao usuário o cancelamento do processo de *upload*. Logo que o *upload* do arquivo é concluído, a barra de progresso localizada à direita do botão *cancel* além de apresentar o progresso como concluído, exibe o nome do arquivo selecionado. Quando isto ocorre, é possível ao usuário prosseguir na seleção dos critérios para a busca.

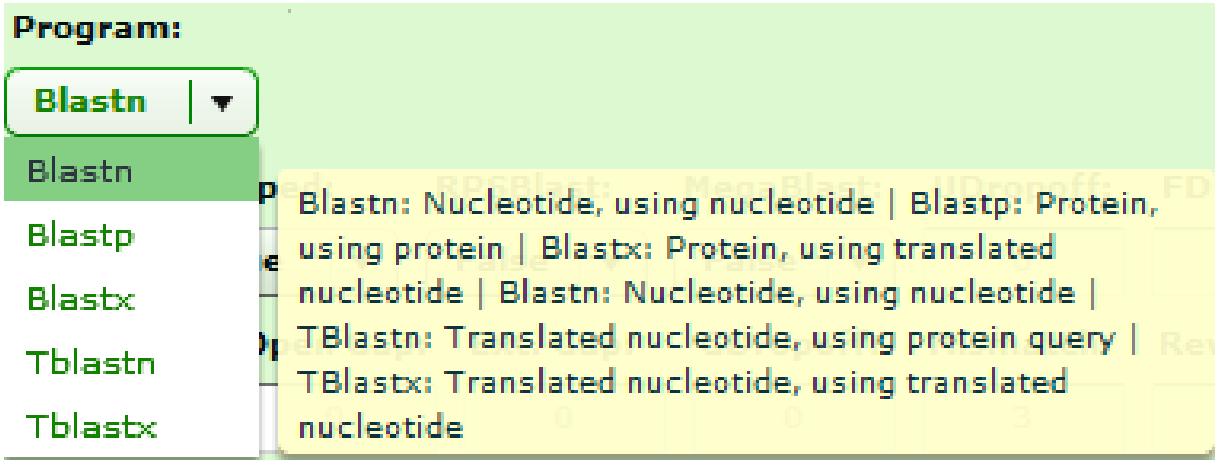


Figura 5.7: Caixa de combinação contendo os tipos de programas disponíveis para a busca e o *tooltip* informativo correspondente.

A caixa de combinação chamada *Program* exibe as opções de programas de busca disponíveis conforme apresenta a Figura 5.7. O primeiro programa de busca disponível é a Blastn, que representa um arquivo de consulta contendo uma sequência de nucleotídeos efetuando uma busca em uma base de dados de nucleotídeos.

O segundo programa de busca disponível é o Blastp que corresponde a um arquivo de consulta contendo proteínas, efetuando uma busca em um banco de dados também de proteínas. O terceiro programa é o Blastx, que corresponde a um arquivo de consulta contendo nucleotídeos traduzidos e a busca é efetuada em um banco de dados de proteínas. O quarto programa disponível é o Tblastn que representa um arquivo de consulta que contém proteínas, efetuando a busca em um banco de dados de nucleotídeos traduzidos. O quinto e último programa disponível é o Tblastx que corresponde a um arquivo de consulta contendo sequências de nucleotídeos traduzidos, efetuando uma busca em um banco de dados também de nucleotídeos traduzidos. Estas opções mencionadas correspondem à opção -**p** da aplicação de linha de comando do Netblast.

Uma vez definido o programa que efetuará a busca no GenBank através do VNBlast, é então possível seguir adiante através da seleção do banco de dados alvo da consulta. Os bancos de dados disponíveis são o *Nucleotide Sequence database, Environmental protein, Environmental nucleotide, EST division database, GSS division database, HTYG division database, Human RefSeq chromosome, Non-Redundant protein database, RefSeq chromosome, Patent protein database, Patent nucleotide database, Protein sequences structures, Nucleotide sequences structures, NCBI genomic reference sequences, NCBI protein reference sequences NCBI transcript reference sequences, STS division database, Swiss-prot sequence databases e, por fim, Whole Genome shotgun*. A caixa de combinação contendo os bancos de dados a

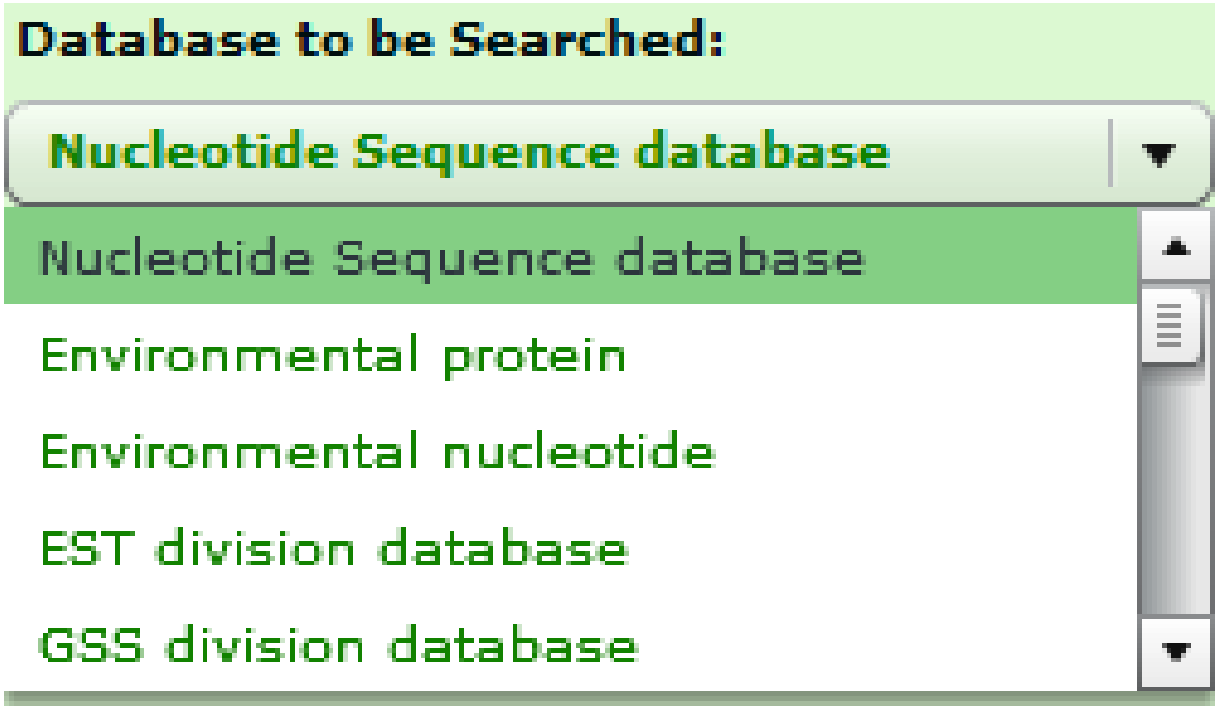


Figura 5.8: Caixa de combinação contendo os tipos de programas disponíveis para a busca.

serem escolhidos pode ser vista na Figura 5.8.

Neste ponto da consulta, todos os parâmetros necessários para uma busca básica baseada nos parâmetros *default* do VNBlast já foram selecionados. Mas dependendo do objetivo da consulta, a forma como os resultados gerados pela busca serão exibidos terá grande impacto no tempo necessário para a obtenção dos resultados esperados pelo pesquisador.

O VNBlast apresenta algumas formas de exibição dos resultados, segundo o mesmo padrão apresentado pelo Netblast. A primeira forma de exibição (também definida como o padrão de visualização de consultas no VNBlast) é o clássico método par-a-par conhecido como *Pairwise*. O segundo formato de visualização é o *Query-anchored showing identities* que significa consultas ancoradas mostrando as identidades entre as sequências. O terceiro formato é o *Query-anchored no identities* que é exatamente idêntico ao segundo formato porém não exibe as identidades entre sequências.

O quarto formato de exibição é similar ao segundo formato, porém exibe os resultados em modo plano. Este formato é chamado de *Flat query-anchored, show identities*. O quinto formato de exibição é similar ao terceiro formato já apresentado, porém, assim como no item anterior, os resultados são apresentados de forma planar. O quinto formato é chamado de *Flat query-anchored, no identities*.

O sexto formato de visualização do alinhamento é chamado de *Query-anchored*

no identities and blunt ends. Este formato é similar ao terceiro porém apresenta cortes mais bruscos nas extremidades dos alinhamentos. O sétimo formato é similar ao sexto formato, porém os resultados são exibidos de forma planar. Este formato é chamado de *Flat query-anchored, no identities and blunt ends*.

O oitavo formato possui características interessantes para a extração dos dados e pós-processamento. Este oitavo formato é o *XML Blast output*. Baseado no formato XML, este formato inclui *tags* informativas nos campos e nos alinhamentos, tornando mais simples a extração dos dados de forma padronizada.

O nono formato é o formato chamado de *Tabular (not post processing)*. Este formato, como o próprio nome diz é apresentado de forma tabular, porém não existem processamentos posteriores ao alinhamento. Este formato pode ser interessante para a extração de sequências, pois os dados são apresentados de forma mais rústica.

O décimo formato disponível é equivalente ao nono, porém recebe alterações de pós-processamento como comentários e ordenação nos alinhamentos, baseados no grau de similaridade entre as sequências encontradas e a sequência de critério. Este formato é chamado de *Tabular with comment lines (post-processed, sorted)*.

O décimo primeiro e último disponível no VNBlast é o formato *ASN, text*. Este formato apresenta os resultados de uma forma textual padrão. Porém um fator muito interessante é a possibilidade de combinação existente entre todos estes formatos disponíveis e a opção HTML, que será mencionada mais adiante. Quando esta opção é selecionada, o resultado exibido de todos estes formatos apresentados vêm acompanhado de *hyperlinks* que possibilitam a navegação através de páginas da internet, para a exibição de detalhes sobre campos mais específicos. A Figura 5.9 apresenta a caixa de combinação do campo Alignment View, que possui 11 opções de visualização dos resultados obtidos na consulta.

Na sequência, é apresentado um grupo de caixas de combinação com funcionalidades variadas, que serão descritas a seguir. A caixa de combinação *Multiple Hit* suporta os parâmetros *Multiple e Single*. O parâmetro *Multiple*, quando selecionado, permite que o alinhamento ocorra através da seleção de múltiplos acertos e o parâmetro *Single* permite apenas a seleção de acerto simples.

A caixa de combinação *HTML* têm um importante papel no VNBlast, pois a sua seleção possibilita a inclusão automática de *tags* do tipo HTML no resultado gerado pelo programa. Uma vez que esta opção esteja marcada como *TRUE*, todos os campos que tenham a possibilidade de exibição de informações adicionais dentro do próprio site do NCBI,

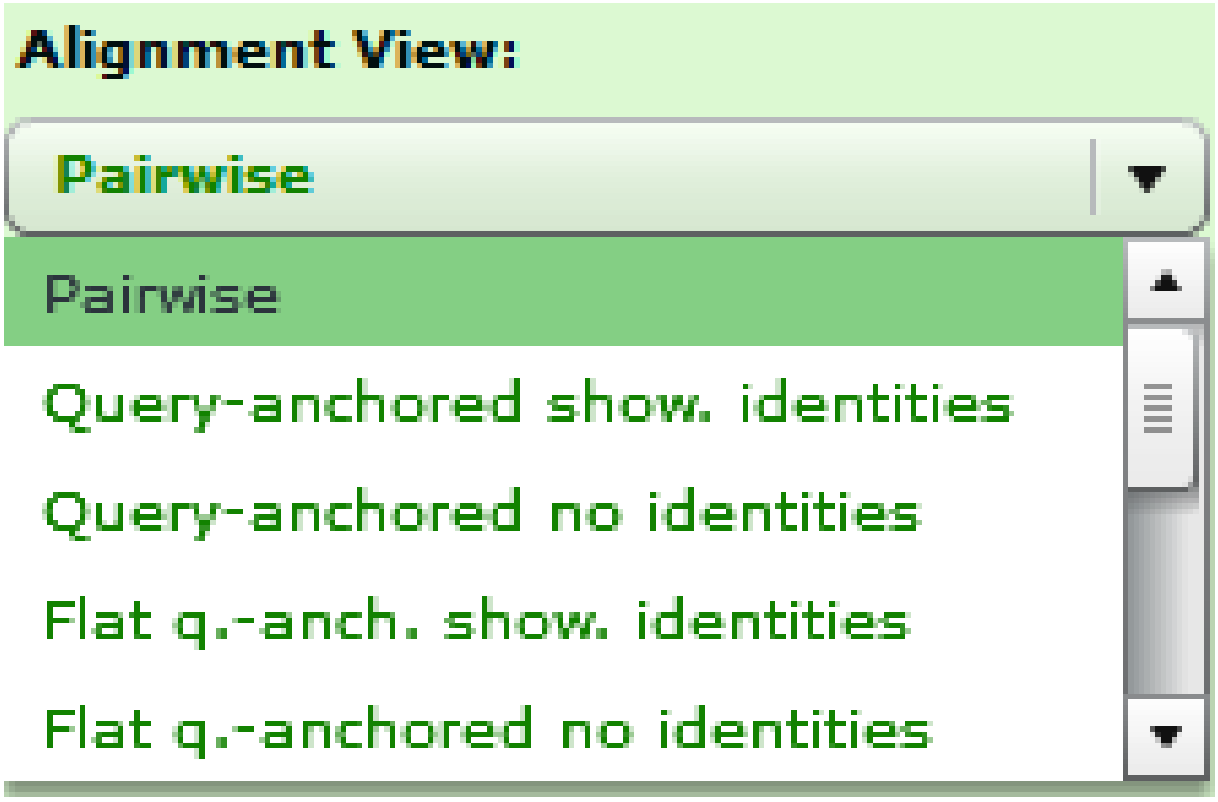


Figura 5.9: Caixa de combinação contendo as opções de formatos de exibição dos resultados no VNBlast.

recebem um *hyperlink* e com um clique do mouse torna-se possível a exibição desta informação adicional dentro da própria janela *Results* do VNBlast, ou através de uma nova janela ou guia do navegador de internet. O valor padrão do VNBlast para o campo HTML é *TRUE* e este parâmetro é equivalente à opção **-T** do Netblast.

O parâmetro *LCase (Lower Case Filtering)* no VNBlast é representado por uma caixa de combinação com os parâmetros *True e False* e é um filtro para bases nitrogenadas representadas por letras em caixa baixa na sequência FASTA. Este parâmetro é equivalente ao parâmetro -U do Netblast e seu valor padrão no VNBlast é *False*.

O próximo parâmetro da sequência é o parâmetro *Believe*. Este parâmetro é equivalente à opção **-J** do Netblast e significa que a aplicação irá confiar na linha de definição da consulta. O valor padrão do VNBlast é *False*, porque a maioria das linhas de definição de consultas não segue o padrão especificado pela NCBI.

A opção *Gapped* do VNBlast é apresentada em uma caixa de combinação que recebe por padrão o valor *True* e sua utilização possibilita a execução de alinhamentos que incluem as lacunas, também conhecidas como (*gaps*) em uma sequência.

O parâmetro *RPSBlast* habilita a opção de busca no banco de dados CDD

Figura 5.10: Grupo de caixas de combinação de variados parâmetros do VNBlast.

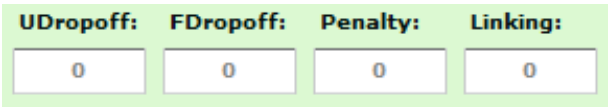


Figura 5.11: Grupo de campos do tipo texto de variados parâmetros do VNBlast.

do GenBank, porém o valor padrão definido no VNBlast é *False* para esta opção. Caso esta opção seja definida como *True*, será necessária a seleção adequada do banco de dados desejado no campo *Database to be Searched*.

A última caixa de combinação deste bloco é a do parâmetro *MegaBlast*. Habilitar esta opção fará com que o VNBlast execute uma busca usando o algoritmo MegaBlast e múltiplas sequências poderão ser concatenadas no critério de consulta. O valor padrão do VNBlast para esta opção é *False*. A Figura 5.10 apresenta a caixa de combinação do campo Alignment View e suas 11 opções de visualização dos resultados obtidos na consulta.

O parâmetro *UDropoff* é a abreviação do parâmetro *X dropoff value for ungapped extensions (in bits)* que significa o valor de deixa X para extensões do tipo *ungapped*. Este parâmetro é equivalente à opção -**y** do Netblast e seu valor padrão no VNBlast é 0.

Já o parâmetro *FDropoff* é a abreviação do parâmetro *X dropoff value for final gapped alignment (in bits)* que significa o valor de deixa para alinhamentos finais do tipo *gapped*. O valor padrão para este parâmetro é 0 e seu parâmetro correspondente dentro do Netblast é o -**Z**.

O parâmetro *Penalty* que corresponde à opção -**q** do Netblast, diz respeito à penalidade aplicada a um erro de comparação entre nucleotídeos e deve ser usada em conjunto com o programa Blastn. Seu uso é indicado para alinhamento de sequências com diferentes percentuais de similaridades.

O parâmetro *Linking* é a representação de *Length of the largest intron allowed in Tblastn for linking HSPs* e é o comprimento do maior *intron* permitido no Tblastn para a ligação de HSPs. O valor padrão 0 do VNBlast desabilita a ligação e qualquer outro valor apresentado ativa este parâmetro. A Figura 5.11 apresenta os campos do tipo *textbox* da segunda linha do VNBlast. Estes campos possuem funções variadas, como descrito anteriormente.

Figura 5.12: Caixa de combinação apresentando os parâmetros suportados pelo campo Mask Filter.

O parâmetro *Mask Filter* foi implementado no VNBlast como uma caixa de combinação para facilitar ao usuário a seleção das opções disponíveis. O objetivo deste parâmetro é o de especificar qual filtro deverá ser usado para mascarar a sequência de consulta. O valor padrão para este campo é *DUST* para nucleotídeos e *SEG* para proteínas. A Figura 5.12 apresenta o campo *Mask Filter* implementado como uma caixa de combinação no VNBlast, assim como as opções suportadas.

Assim como o parâmetro *Mask Filter* e outros, o parâmetro *P. Scoring* também possui um conjunto limitado de parâmetros suportados e foi implementado através de uma caixa de combinação. Este campo está relacionado com a matriz proteica de pontuação para uso. Os parâmetros suportados são o BLOSUM 45, BLOSUM62, BLOSUM 80, PAM30 e o PAM70. O VNBlast utiliza como valor padrão para este campo o parâmetro BLOSUM62. A Figura 5.13 apresenta o campo *Mask Filter* implementado também como uma caixa de combinação no VNBlast, assim como os devidos parâmetros suportados.

O campo *Cutoff* receberá o valor de corte a ser utilizado pelo algoritmo BLAST. A alteração do valor padrão deste campo fará com que os resultados sejam drasticamente alterados, dependendo do valor de corte especificado. Este campo só deve ser alterado caso o usuário realmente saiba em quê implicará esta alteração, caso contrário resultados

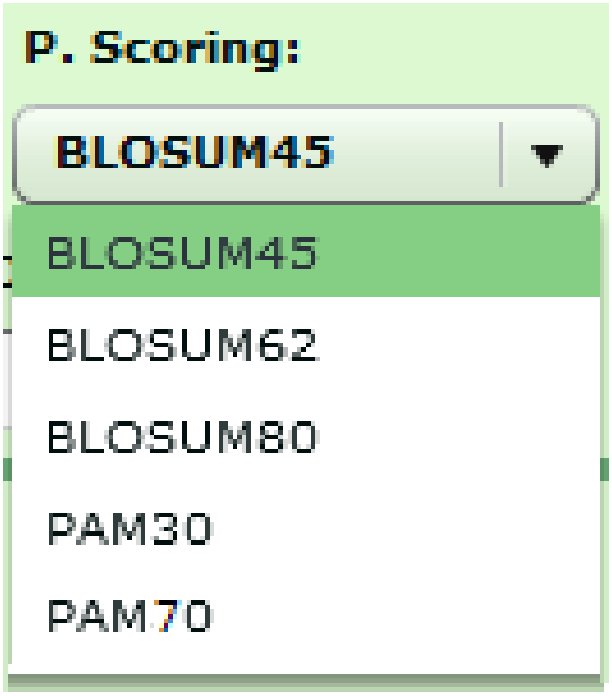


Figura 5.13: Caixa de combinação apresentando os parâmetros suportados pelo campo P. Scoring.

incorretos poderão ser gerados. Este campo suporta alguns formatos de parâmetros, como números inteiros, fracionários, decimais, exponenciais além de números em notação científica. O valor estipulado como padrão para este campo pelo VNBlast é o número 10. O parâmetro equivalente a este pelo Netblast é a opção -**e**.

O parâmetro *Open Gap* representa o custo ou penalidade para a abertura de um *gap* pelo VNBlast. A alteração do valor deste parâmetro fará com que seja alterado o custo de abertura de um *gap*. O valor padrão para este parâmetro pelo VNBlast é 0 e o parâmetro equivalente a este no Netblast seria a opção -**G**.

De modo similar, o parâmetro *Ext. Gap* representa o custo ou penalidade para a extensão de um *gap* pelo VNBlast. A alteração do valor deste parâmetro fará com que seja alterada a penalidade pela extensão de um *gap*. O valor padrão para este parâmetro pelo VNBlast é 0 e o parâmetro equivalente a este no Netblast seria a opção -**E**.

No VNBlast, o parâmetro *GDropoff* significa o valor de deixa X para alinhamentos do tipo *gapped* e os valores aceitos são apresentados em *bits*. Por exemplo, para se alterar o valor de deixa de um alinhamento do tipo *gapped* para 40, é só atribuir o valor 40 a este campo. O valor padrão atribuído pelo VNBlast para este campo é 0 e a opção equivalente a este parâmetro pelo Netblast seria a -**X**.

A penalização para os erros em um alinhamento de nucleotídeos no VNBlast

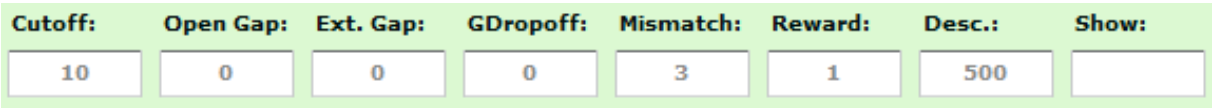


Figura 5.14: Campos diferenciados com funcionalidades variadas.

é definida pelo valor do campo *Mismatch*. Este parâmetro só será avaliado caso o programa escolhido seja o Blastn, específico para nucleotídeos. O valor definido como padrão para este parâmetro é o -3, mas a alteração para outros valores pode ser útil quando se está efetuando o alinhamento de sequências com diferentes percentuais de similaridade.

De modo oposto à penalização especificada no item anterior, o campo *Reward* representa a recompensa pelo acerto em um nucleotídeo. Uma alternativa a esta definição manual seria o uso de uma matriz de pontuação externa. O valor padrão definido no VNBlast para este campo é o número 1.

O campo abreviado como *Desc.* no VNBlast significa descrição e especifica o limite superior do banco de dados de sequências cujas descrições serão exibidas. O valor padrão para o número de sequências que receberão a descrição é 500, mas este valor poderá ser aumentado ou reduzido conforme a necessidade do solicitante. O parâmetro do Netblast equivalente a este é o **-v**.

O parâmetro *Show* especifica o limite superior das sequências do banco de dados cujos alinhamentos serão mostrados no resultado pelo VNBlast. O limite físico para este parâmetro é 200.000, porém este valor não representa o número total de segmentos de alinhamento nem os *high scoring pairs (HSPs)*. Ao invés disso, é o número de sequências do banco de dados com os HSPs para a consulta. O valor padrão deste parâmetro é 250 e a opção equivalente do Netblast seria a **-b**. A Figura 5.14 apresenta os campos mencionados no VNBlast, assim como os devidos parâmetros suportados.

A próxima caixa de combinação é a *Show GI* e como o próprio nome sugere, a função deste campo é ativar e desativar a adição do número GI na linha de definição do resultado. Quando esta opção é habilitada, o número GI precederá o identificador PRF na linha da sequência. O valor padrão para esta opção é desativado *(False)* e este parâmetro corresponde à opção **-I** do Netblast.

A abreviação *Gen. Cod.* que corresponde a *Query genetic code to use*, representa o código genético da consulta a ser usado. Isto especifica a tabela de tradução usada na tradução da consulta durante buscas utilizando os programas Blastx e Tblastx. O valor padrão 1 adotado pelo VNBlast corresponde à codificação universal. O parâmetro

equivalente usado no Netblast é o -**Q**.

O parâmetro *DB Gen.*, exibido no VNBlast, corresponde à *DB Genetic code*, ou seja, o código genético do banco de dados. Assim como a opção anterior, esta opção é utilizada para para informar a tabela de tradução que será utilizada na tradução do banco de dados em buscas usando os programas Tblastn e Tblastx. O valor padrão para este campo é 1.

Na sequência, o parâmetro *Wd. Size* do VNBlast define, como a própria abreviação sugere, o tamanho das palavras usadas nos variados programas da família Blast. O valor padrão 0, define implicitamente valores padrão para alguns programas. Por exemplo, o valor padrão para o programa Blastn é 11, para o Megablast, o valor é 28, já para todos os demais, o valor padrão para este campo é 3. Alterando o valor deste campo, o VNBlast compreenderá que o valor definido será correspondente ao programa selecionado no momento da consulta.

Já o parâmetro *Lenght* do VNBlast é usado para definir o tamanho efetivo do banco de dados. Apesar de que o valor padrão do VNBlast para este campo é 0, este valor é interpretado pelo VNBlast como o tamanho atual do banco de dados, sem restrições. No Netblast, a opção correspondente é definida através do parâmetro -**z**.

O parâmetro *Best Hits*, que corresponde à opção -K do Netblast, significa o número dos melhores acertos encontrados em uma região conservada. Definir este parâmetro faz com que o VNBlast selecione o número especificado com os melhores acertos para uma dada região de uma consulta para avaliação. O valor padrão desta opção é 0, mas caso seja necessário, o valor 100 é recomendado.

O parâmetro *E. Lenght*, que é a simplificação da palavra *Efective Lenght*, significa o tamanho efetivo do espaço de busca. Este parâmetro é definido pelo produto do tamanho efetivo da consulta menos o tamanho atual corrigido para efeitos de borda. Para se utilizar o tamanho atual, deve-se manter o valor padrão 0 do VNBlast.

O parâmetro *Strands* do VNBlast significa as vertentes da consulta de nucleotídeos para serem usadas na busca. Alguns valores são utilizados por este campo: o valor 1 significa a sequência de entrada. O valor 2 significa o complemento reverso e o valor padrão 3 significa ambas as situações. A opção correspondente a esta no Netblast é a -**S**.

No caso do parâmetro *Subset* do VNBlast, seu objetivo é restringir a busca do banco de dados ao subconjunto que satisfaz a consulta. Os argumentos para este campo são o conjunto de parâmetros válidos pré-definidos pelo Entrez. O servidor Blast utilizará

Figura 5.15: Campos diferenciados com funcionalidades variadas.

Figura 5.16: Barra de progresso exibida através da solicitação de uma consulta no VNBlast.

este termo para recuperar a lista de números GI e usá-los para restringir a busca do Blast às entradas especificadas nesta lista. Apenas termos válidos devem ser usados. Por exemplo, não fará sentido restringir uma busca para sequência genômicas quando a busca está sendo feita em um banco de dados do tipo EST.

O VNBlast possui um campo chamado *Location*. Este campo está relacionado com a localização em uma sequência de consulta. Por exemplo, quando se define os valores 100,400 , o valor 100 representa o início e o valor 400 representa o fim. Este parâmetro corresponde à opção -**L** do Netblast.

O parâmetro *Win. Size* que é a abreviação da palavra *window size* representa o tamanho da janela para múltiplos acertos. O valor padrão para este campo é 0, porém o real valor quando se define este campo com 0 é 0 para o Blastn e Megablast e 40 para os demais programas do pacote Blast. Este parâmetro é correspondente à opção -**A** do Netblast.

Por fim, o parâmetro *Threshold*, significa o limite para acertos estendidos. O valor padrão 0 do VNBlast define valores diferenciados para os variados programas do pacote Blast que são: Blastp = 11, Blastn = 0, Blastx = 12, Tblastn = 13, Tblastx = 13 e Megablast = 0. Este parâmetro é correspondente à opção -**f** do Netblast. A Figura 5.15 apresenta os campos mencionados no VNBlast, assim como os devidos parâmetros suportados.

No momento em que uma consulta é solicitada, surge uma barra de progresso do tipo infinito no VNBlast, informando ao usuário que a consulta está em andamento e que será necessária a espera pela conclusão do procedimento. A Figura 5.16 apresenta a barra de progresso para uma consulta no VNBlast.

O VNBlast apresenta alguns botões de comando que têm por função a execução de procedimentos relativos à buscas, campos do formulário e ao próprio formulário. Estes botões receberão uma descrição mais detalhada nos parágrafos que seguem.

O botão Search, como o próprio nome sugere, é o botão que, quando pressionado, informa ao VNBlast que o conjunto de informações compostos pelo arquivo contendo a sequência de critério para o alinhamento e os próprio parâmetros do VNBlast selecionados

Figura 5.17: Conjunto de botões com funcionalidades variadas no formulário de consultas do VNBlast.

pelo usuário deverão ser submetidos ao GenBank para a execução dos procedimentos de busca e alinhamento definidos na consulta.

Por padrão, o botão *Search* do VNBlast permanece desabilitado, enquanto um conjunto mínimo de informações necessárias para a execução de uma consulta não são informados pelo usuário. Uma vez que estes requisitos são preenchidos então o botão Search é habilitado automaticamente, tornando então possível a execução da busca e alinhamento desejados.

O segundo botão localizado na parte inferior do formulário de consultas é o botão *Clear*. Este botão têm a função de limpar todos os campos do VNBlast, incluindo o arquivo de critério, caso o mesmo já tenha sido selecionado anteriormente. Este botão é útil quando é necessária a execução de várias buscas sucessivas ou quando algum arquivo de critério ou conjunto de parâmetros foi informado de forma equivocada. Os campos de parâmetros do VNBlast também são reiniciados com os seus respectivos valores padrão.

O próximo botão da esquerda para a direita é o botão *Default*, também localizado na parte inferior do formulário de consultas. Este botão, assim como o *Clear*, também restaura os parâmetros aos seus devidos valores iniciais, porém, caso o arquivo de critério já tenha sido selecionado, o mesmo continuará sendo mantido de forma que apenas os parâmetros de busca sejam alterados. Este botão é útil quando se deseja variar os parâmetros de busca e verificar os resultados de forma sucessiva.

O quarto botão na sequência é o botão *Download*. Este botão, quando pressionado, possibilita ao usuário efetuar a baixa do arquivo resultante de uma consulta, possibilitando o salvamento de modo definitivo e local dos resultados do VNBlast. Este botão permanece desabilitado até que o resultado de uma busca seja exibido no VNBlast, sendo então automaticamente habilitado.

Por fim o botão Close, que fica localizado no canto inferior direito do formulário de consultas do VNBlast, têm por função o fechamento apenas do formulário de consultas, sendo possível a reabertura deste formulário através do menu do formulário principal do VNBlast. A Figura 5.17 apresenta o conjunto de botões com funcionalidades variadas no formulário de consultas do VNBlast.

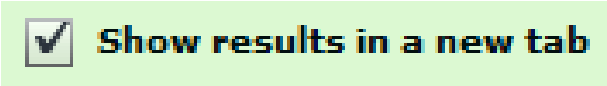


Figura 5.18: Caixa de checagem *Show results in a new tab* do formulário de consultas do VNBlast.

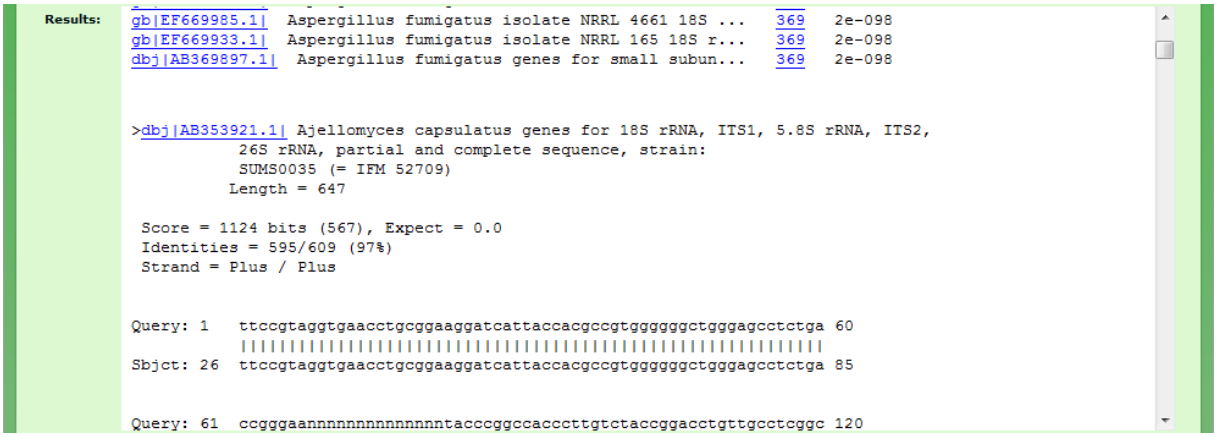


Figura 5.19: Campo Results do formulário de consultas do VNBlast. Este campo exibe os resultados das consultas geradas pelo VNBlast.

Uma vez que os parâmetros mínimos necessários para a execução de uma consulta são preenchidos, então o VNBlast está apto a efetuar a busca por sequências similares nas bases de dados do GenBank. Quando o processo é concluído, então o resultado é gerado e retornado ao servidor do VNBlast, que por sua vez, transfere os dados para o cliente.

Quando isso acontece, os dados resultantes da consulta são exibidos no campo *Results* do VNBlast ou no campo *Results* do VNBlast e em uma nova aba do navegador em que o VNBlast esteja sendo executado.

A seleção do modo de visualização é feita através da caixa de checagem chamada *Show results in a new tab*. Quando desmarcada, os resultados serão exibidos apenas na janela *Results* do formulário de consultas do VNBlast. Caso esta opção esteja marcada, então os resultados serão exibidos tanto na janela *Results* do VNBlast quanto em uma nova guia do navegador corrente e no momento em que os resultados são gerados, o foco da aplicação é transferido para a nova guia, porém a guia que contém a aplicação do VNBlast não é encerrada. A Figura 5.18 apresenta a caixa de checagem *Show results in a new tab* do formulário de consultas do VNBlast. A Figura 5.19 apresenta a janela *Results* do formulário de consultas do VNBlast.

A Figura 5.20 apresenta os resultados da consulta sendo exibidos em uma nova guia do navegador de internet corrente.

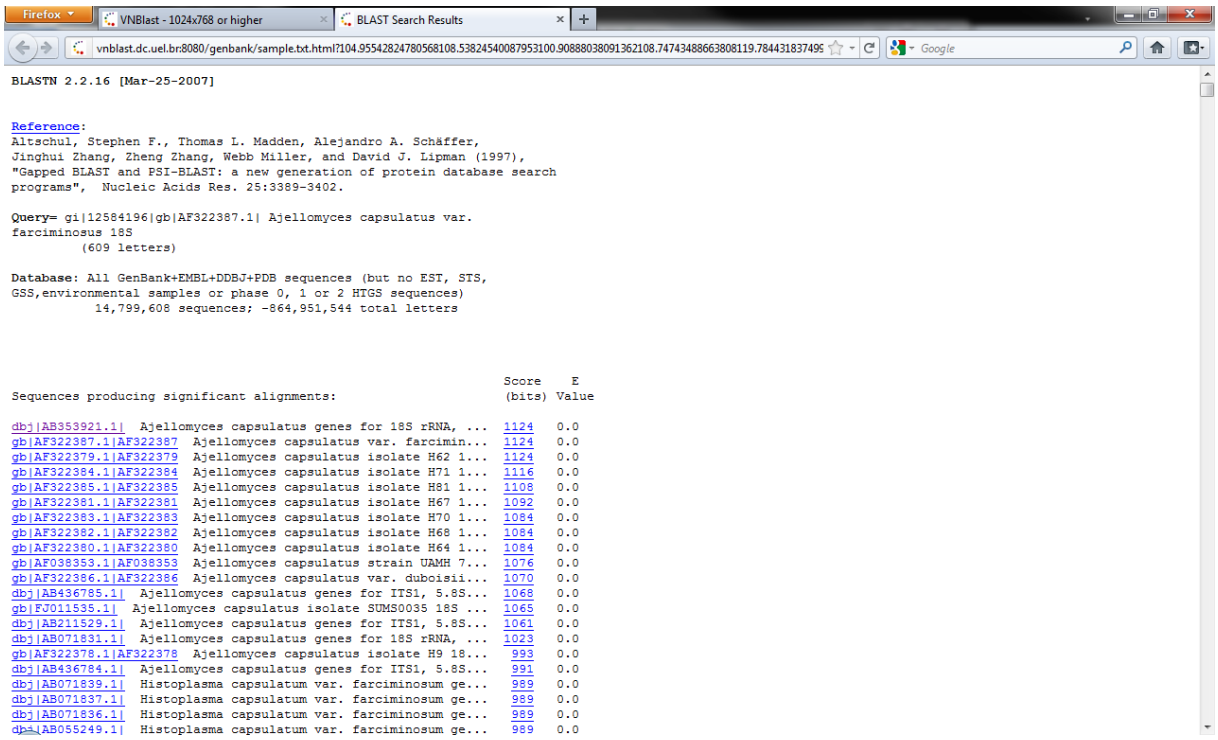


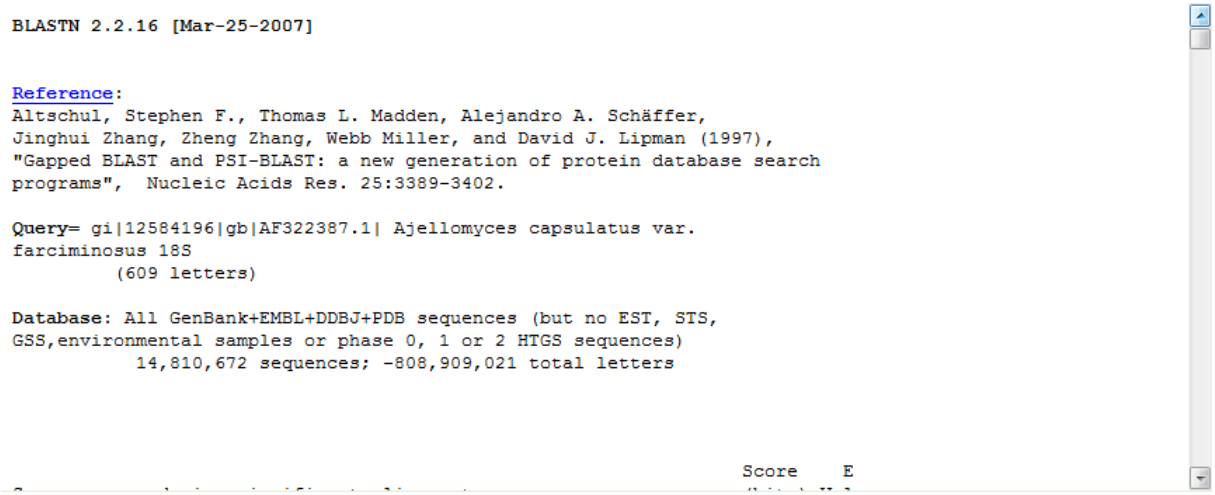
Figura 5.20: Resultados da consulta sendo exibidos em uma nova guia do browser corrente.

5.1 Análise dos Resultados

Os formatos de saída gerados pelo VNBlast podem sofrer várias alterações dependendo do conjunto de parâmetros definidos pelo usuário. Dentre estes, de modo enfático, o parâmetro *Alignment View* tem a capacidade de alterar significativamente o modo como os dados são exibidos.

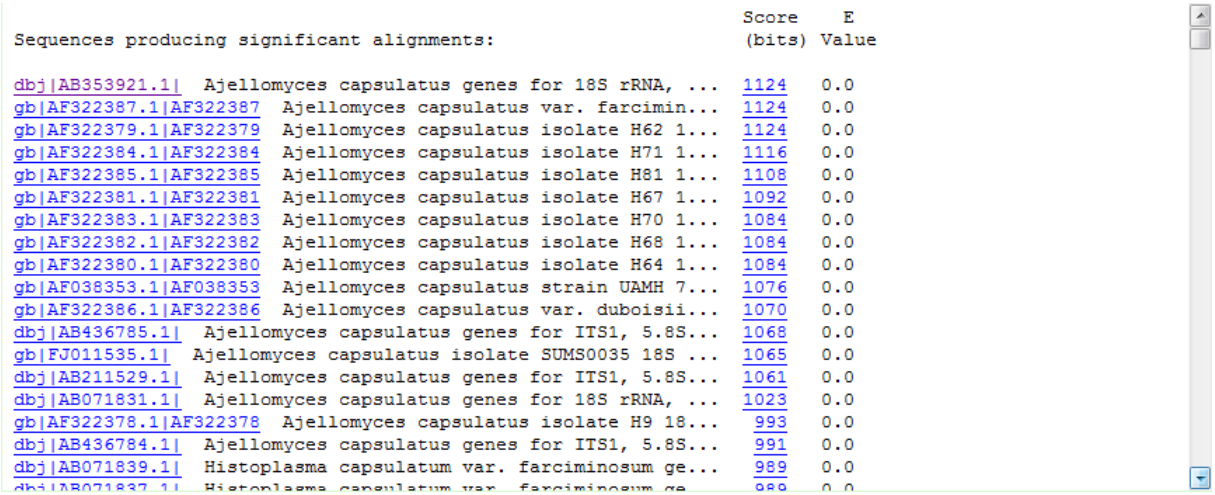
Um dos modos mais utilizados para visualização dos resultados é o par-a-par conhecido como *Pairwise*. Desta forma, os exemplos utilizados para a análise dos resultados serão feitos baseados no modo *Pairwise*. Contudo, vale lembrar que este modo corresponde a apenas um dos onze modos de vizualização disponíveis no VNBlast.

Quando um resultado é gerado no VNBlast no modo *Pairwise*, alguns blocos de conteúdos bem definidos são criados. A primeira parte do resultado gerado pelo VNBlast contém informações sobre a versão do programa interno usado pela NCBI para a busca e alinhamento. Na sequência é feita uma referência bibliográfica aos autores do algoritmo BLAST utilizado em todo o procedimento. O próximo campo apresenta o cabeçalho da sequência submetida como critério no formato FASTA. Por fim, neste primeiro bloco são apresentados os bancos de dados atualmente interconectados e disponíveis para consulta através do GenBank e algumas informações estatísticas sobre as sequências pesquisadas. A Figura 5.21 apresenta



```
BLASTN 2.2.16 [Mar-25-2007]


Reference:
Altschul, Stephen F., Thomas L. Madden, Alejandro A. Schäffer,
Jinghui Zhang, Zheng Zhang, Webb Miller, and David J. Lipman (1997),
"Gapped BLAST and PSI-BLAST: a new generation of protein database search
programs",  Nucleic Acids Res. 25:3389-3402.

Query= gi|12584196|gb|AF322387.1| Ajellomyces capsulatus var.
farciminosus 18S
         (609 letters)

Database: All GenBank+EMBL+DDBJ+PDB sequences (but no EST, STS,
GSS,environmental samples or phase 0, 1 or 2 HTGS sequences)
           14,810,672 sequences; -808,909,021 total letters




                                                                  Score    E
```

Figura 5.21: A primeira parte dos resultados gerados por uma consulta através do VNBlast no formato par-a-par.



Sequences producing significant alignments:		Score (bits)	E Value
dbj|AB353921.1|	Ajellomyces capsulatus genes for 18S rRNA, ...	1124	0.0
gb|AF322387.1|AF322387	Ajellomyces capsulatus var. farcimin...	1124	0.0
gb|AF322379.1|AF322379	Ajellomyces capsulatus isolate H62 1...	1124	0.0
gb|AF322384.1|AF322384	Ajellomyces capsulatus isolate H71 1...	1116	0.0
gb|AF322385.1|AF322385	Ajellomyces capsulatus isolate H81 1...	1108	0.0
gb|AF322381.1|AF322381	Ajellomyces capsulatus isolate H67 1...	1092	0.0
gb|AF322383.1|AF322383	Ajellomyces capsulatus isolate H70 1...	1084	0.0
gb|AF322382.1|AF322382	Ajellomyces capsulatus isolate H68 1...	1084	0.0
gb|AF322380.1|AF322380	Ajellomyces capsulatus isolate H64 1...	1084	0.0
gb|AF038353.1|AF038353	Ajellomyces capsulatus strain UAMH 7...	1076	0.0
gb|AF322386.1|AF322386	Ajellomyces capsulatus var. duboisii...	1070	0.0
dbj|AB436785.1|	Ajellomyces capsulatus genes for ITS1, 5.8S...	1068	0.0
gb|FJ011535.1|	Ajellomyces capsulatus isolate SUMS0035 18S ...	1065	0.0
dbj|AB211529.1|	Ajellomyces capsulatus genes for ITS1, 5.8S...	1061	0.0
dbj|AB071831.1|	Ajellomyces capsulatus genes for 18S rRNA, ...	1023	0.0
gb|AF322378.1|AF322378	Ajellomyces capsulatus isolate H9 18...	993	0.0
dbj|AB436784.1|	Ajellomyces capsulatus genes for ITS1, 5.8S...	991	0.0
dbj|AB071839.1|	Histoplasma capsulatum var. farciminosum ge...	989	0.0

Figura 5.22: A segunda parte dos resultados gerados por uma consulta através do VNBlast no formato par-a-par.

a primeira parte dos resultados gerados por uma consulta através do VNBlast sendo exibidos no formato par-a-par.

No segundo bloco de conteúdos gerados como resultado de uma consulta no modo *Pairwise* pelo VNBlast, é apresentado para cada sequência similar encontrada no banco de dados em que a sequência está armazenada, o índice de identificação, o nome que foi dado à sequência submetida, a pontuação gerada pelo algoritmo BLAST (*score*) em relação à similaridade desta em comparação com a sequência da consulta e, da mesma forma, a margem de erro dada pelo BLAST pela comparação entre as sequências. A Figura 5.22 apresenta a segunda parte dos resultados gerados por uma consulta através do VNBlast sendo exibidos no formato par-a-par.

O terceiro bloco de conteúdos obtidos exibe novamente o cabeçalho da

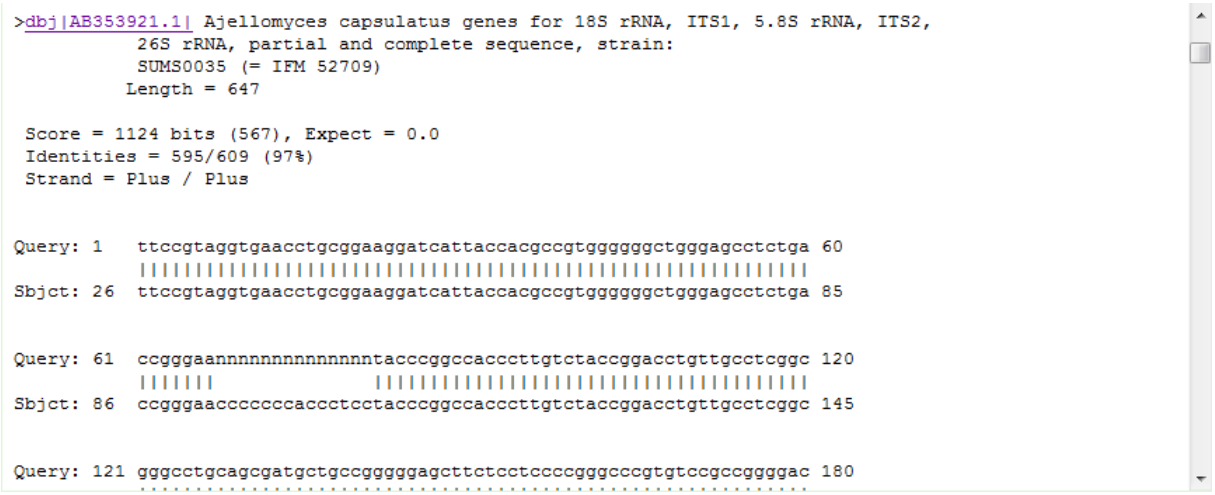


Figura 5.23: A terceira parte dos resultados gerados por uma consulta através do VNBlast no formato par-a-par.

sequência encontrada como similar pelo algoritmo BLAST, um índice de pontuações de identidades obtidas e por fim o próprio alinhamento entre blocos das sequências de critério encontradas no banco. O padrão do VNBlast é exibir os alinhamentos das sequências em blocos de 60 em 60 pares de bases nitrogenadas. A Figura 5.23 apresenta a terceira parte dos resultados gerados por uma consulta através do VNBlast sendo exibidos no formato par-a-par.

Em uma análise mais detalhada neste exemplo de alinhamento do tipo par-a-par, é possível encontrar os 3 tipos básicos de ocorrências em um alinhamento deste tipo: O acerto (*match*), o erro (*mismatch*) e a lacuna (*gap*). O acerto ocorre quando as bases nitrogenadas alinhadas são iguais. O erro acontece quando as bases nitrogenadas alinhadas são diferentes. Por fim, a lacuna ocorre quando em um alinhamento, uma das sequências não encontra uma base nitrogenada na outra sequência, apenas uma lacuna. A Figura 5.24 apresenta exemplos de acertos (*match*) e erros (*mismatch*) gerados por uma consulta através do VNBlast sendo exibidos no formato par-a-par. Já a Figura 5.25 apresenta um exemplo de lacuna (*gap*) gerada por uma consulta através do VNBlast no formato par-a-par.

Ao final da consulta, é exibido um bloco contendo informações estatísticas e de resumo dos resultados obtidos em uma determinada consulta. Estas informações podem ser úteis para uma análise geral dos resultados obtidos. A Figura 5.26 apresenta o último bloco de informações gerado por uma consulta através do VNBlast, quando exibida no formato par-a-par.

Segundo as últimas figuras apresentadas, é possível perceber que vários itens exibidos na janela *Results* do VNBlast possuem *hyperlinks*. Estes *hyperlinks* apontam para outras páginas dentro do domínio do NCBI, que possuem informações complementares ao item

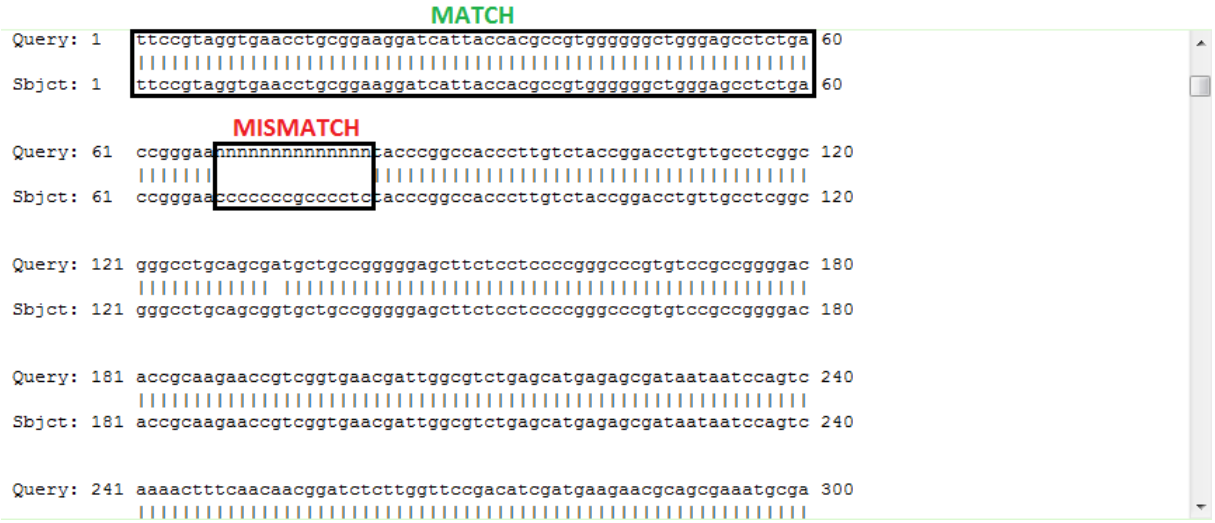


Figura 5.24: Exemplos de acertos (*match*) e erros (*mismatch*) gerados por uma consulta através do VNBlast no formato par-a-par.

Figura 5.25: Exemplo de lacuna (*gap*) gerada por uma consulta através do VNBlast quando exibida no formato par-a-par.

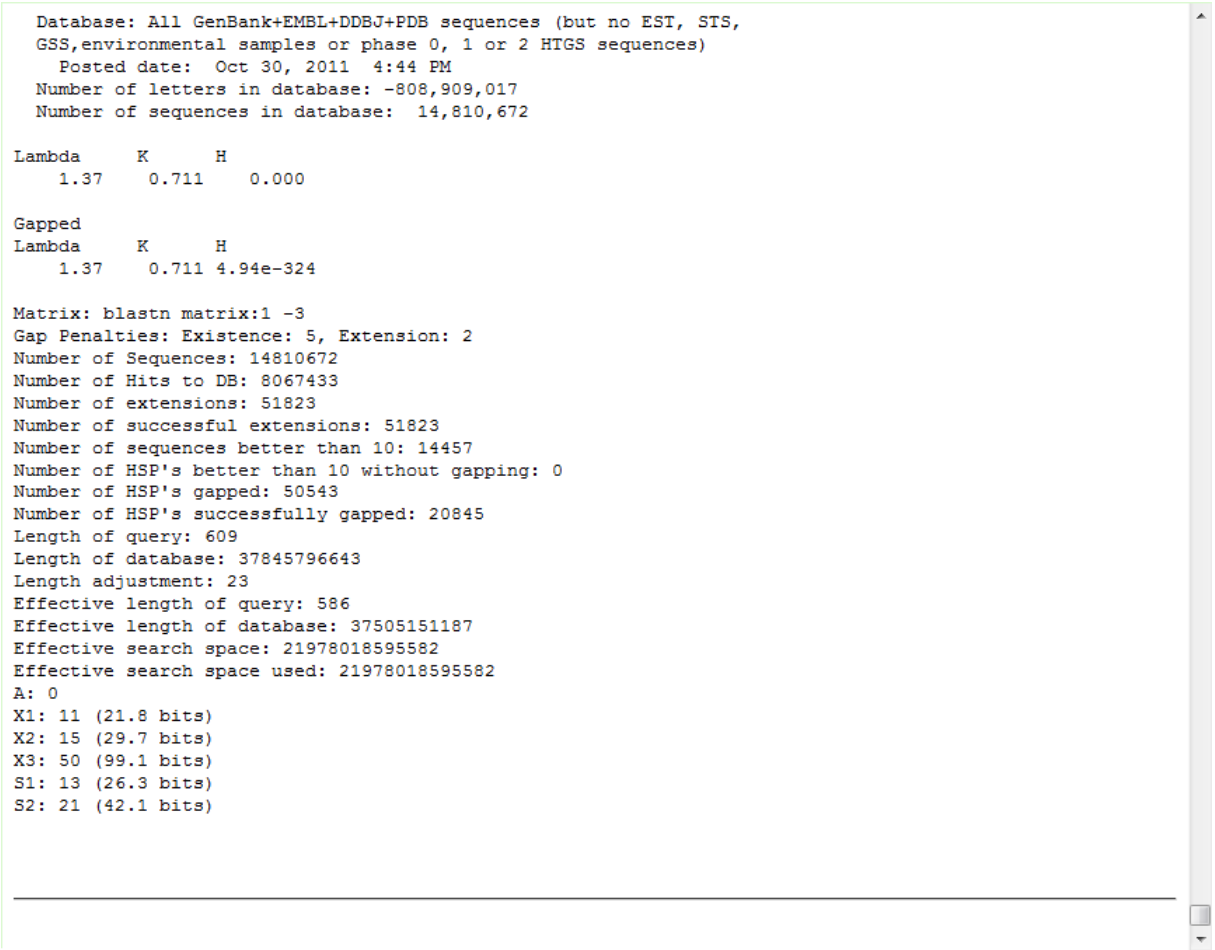


Figura 5.26: Último bloco de informações gerado por uma consulta através do VNBlast no formato par-a-par.

relacionado, exibido no VNBlast.

O VNBlast por sua vez, possibilita que páginas HTML da internet sejam exibidas internamente através da janela *Results*. Desta forma, não é necessária a cópia do endereço a que o *hyperlink* aponta, para a posterior execução do mesmo em um navegador de internet, uma vez que é possível a exibição da página na própria janela *Results* do VNBlast, bastando para isso um simples clique do mouse sobre o *hyperlink* desejado.

O fato de que o VNBlast permite a exibição de páginas HTML auxilia os pesquisadores na obtenção de informações externas ao BLAST, como por exemplo a localização de sequências originais em formato FASTA de organismos encontrados como similares em uma consulta no VNBlast. Além disso, o site da NCBI possue uma grande biblioteca de artigos científicos que podem ser acessados através de *hyperlinks* no VNBlast.

Além do modo de exibição embutida no VNBlast, os mesmos resultados podem ser exibidos em uma nova guia do navegador utilizado pelo usuário, permitindo uma melhor vizualização dos resultados e a mesma possibilidade de navegação através dos *hyperlinks* gerados pela consulta. A Figura 5.27 apresenta o exemplo de uma página da NCBI apontada por um *hyperlink* da consulta gerada pelo VNBlast, sendo exibida dentro do próprio VNBlast e contendo informações específicas sobre um determinado organismo apresentado na consulta.

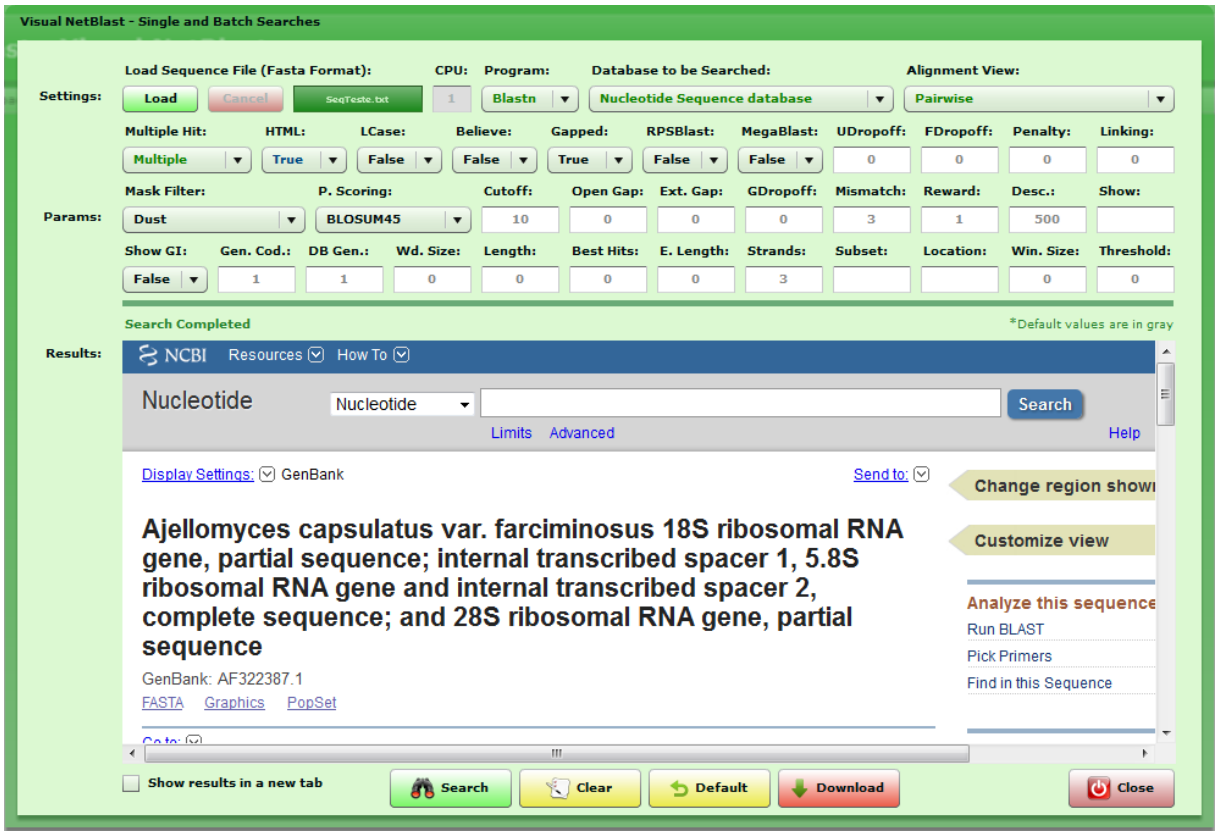


Figura 5.27: Exemplo de uma página da NCBI apontada por um *hyperlink* na consulta gerada pelo VNBlast sendo exibida dentro do próprio VNBlast.

6 Conclusão e Trabalhos Futuros

Este trabalho teve como objetivo o desenvolvimento de um mecanismo que pudesse compensar as limitações existentes na atual página do NCBI-BLAST, relacionadas ao número de parâmetros de busca e alinhamento disponíveis e a impossibilidade de buscas em lote. Quando em comparação com outras ferramentas descritas neste trabalho, o principal objetivo era o de possibilitar variadas opções de pré-processamento e possibilitar ao usuário a obtenção de informações sempre atualizadas do GenBank.

Além destes objetivos terem sido alcançados no desenvolvimento deste trabalho, o VNBlast se mostrou uma interface amigável com o usuário, possibilitando que as etapas necessárias para a execução de uma busca fossem efetuadas de forma intuitíva e acessível, minimizando o tempo necessário para a obtenção dos resultados finais pelo pesquisador.

O VNBlast possibilitou a obtenção de informações externas ao tradicional método de alinhamento de forma simplificada através da navegação entre páginas HTML, através do uso de *hyperlinks* dentro do próprio formulário do VNBlast, ampliando suas possibilidades de utilização.

Em adição às características mencionadas que foram alcançadas com o desenvolvimento deste programa, o VNBlast possibilitou ao usuário o salvamento definitivo dos resultados na forma de um arquivo HTML ou texto. Caso o arquivo tenha sido salvo no formato HTML, então será possível o acesso às informações externas ao próprio arquivo através dos *hyperlinks* que ficarão salvos neste arquivo, bastando para isso que exista apenas uma conexão de internet disponível.

6.1 Contribuições

Como contribuição destre trabalho é possível destacar o surgimento desta nova ferramenta, VNBlast, que facilitará o processo de buscas e alinhamentos de uma forma visual e intuitiva para pesquisadores e estudantes. O VNBlast não necessita de prévia instalação

local para sua execução, sendo necessário apenas um navegador de internet e o plug-in do Flash préviamente instalado. Os dados gerados pelo VNBlast estão sempre atualizados e existe a possibilidade de execução de buscas em lote.

No desenvolvimento deste projeto foram utilizadas tecnologias modernas e como contribuição para a comunidade científica, será disponibilizado o código fonte deste projeto para download, de forma que outras ferramentas ou variações do VNBlast possam surgir.

6.1.1 Publicações

As seguintes publicações estão associadas a este trabalho:

- VNBlast: A Netblast Management System,

  Marcio Rodrigo Santos, Wesley Attrot and Emerson José Venâncio,

  IADIS - International Association for Development of the Information Society.

  ISBN: 978-989-8533-06-7.

  Páginas (461 - 464).

  07 de Novembro de 2011.

  Rio de Janeiro, Brazil.

- Acesso, Busca e Alinhamento de Sequências Genéticas ao Banco de Dados GenBank,

  Marcio Rodrigo Santos, Wesley Attrot, Alan Salvany Felinto e Jacques Duílio Brancher,

  CICPG - Congresso de Iniciação Científica e Pós-Graduação - Sul Brasil.

  Setembro de 2010.

  Florianópolis, Brazil.

6.2 Trabalhos Futuros

Em adição aos recursos atualmente disponíveis no VNBlast, novas funcionalidades estão sendo adicionadas, como a inclusão de um banco de dados para o armazenamento e manipulação dos resultados gerados e a inclusão de um novo mecanismo de alinhamento global de sequências genéticas. Outra funcionalidade, atualmente em desenvolvimento, é a

extração das sequências resultantes das consultas, para a conversão destas ao formato FASTA, permitindo a retroalimentação dos dados em novas consultas.

Referências Bibliográficas

ADACHI, J.; HASEGAWA, M. Molphy version 2.3 - programs for molecular phylogenetics based on maximum likelihood. *Technical Report*, 1996.

ADOBE. *Adobe Flex*. November 2011. Disponível em: <http://www.adobe.com/br-/products/flex/>.

AL., J. C. et. *OCGC Blast*. November 2011. Disponível em: <http://www.ocgc.ca/ocgcblast-.htm>.

ALTSCHUL, S. F. et al. Basic local alignment search tool. *J. Mol. Biol. (1990).*, v. 215, p. 403–410, 1990.

ALTSCHUL, S. F. et al. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research (1997).*, v. 25, p. 3389–3402, 1997.

ALTSCHUL, S. K. e S. F. Karlin e altschul. *Proc. Nat. Acad. Sci. U.S.A.*, v. 87, p. 2264–2268, 1990.

AMARAL, J. B. do; SANTELLI, G. M. M. A cultura de células em 3 dimensões e a sua aplicação em estudos relacionados a formação do lúmen. *Naturalia (2011).*, v. 34, p. 1–20, 2011.

BALLY, P. et al. Fonzie: An optimized pipeline for minisatellite marker discovery and primer design from large sequence data sets. *BioMed Central*, v. 3, p. 322–325, 2010.

BENSON, D. et al. Genbank. *Nucleic Acids Research (2010).*, v. 38, 2009.

BERMAN, H. M. et al. The protein data bank. *Nucleic Acids Ressearch*, v. 28, p. 235–242, 2000.

BL, M. et al. The rdp-ii (ribosomal database project). *Nucleic Acids Research*, v. 29, p. 173–174, 2001.

BOONE, M.; UPTON, C. Blast search updater: a notification system for new databases matches. *Bioinformatics Applications Note (2000).*, v. 16, p. 1054–1055, 2000.

COMMUNITY, B. *BioPerl*. November 2011. Disponível em: <http://www.bioperl.org/>.

COMUNITY, J. *JBoss*. November 2011. Disponível em: <http://www.jboss.org/>.

DAYHOFF, R. M. S. e. B. C. O. M. O. In atlas of protein sequence and structure. *Nat. Biomed. Res. Found.*, v. 5, p. 345–352, 1978.

DIENER, S. E. et al. Alkahest nuclearblast : a user-friendly blast management and analysis system. *BMC Bioinformatics (2005).*, v. 6, p. 147–153, 2005.

EDDY, S. R. A probabilistic model of local sequence alignment that simplifies statistical significance estimation. *PloS Computational Byology*, v. 4, 2008.

FARMERIE, W. G. et al. Biological workflow with blastquest. *Data and Knowledge Engineering (2005).*, v. 53, p. 75–97, 2004.

FELSENSTEIN, J. Phylip-phylogeny inference package (version 3.2). *Cladistics*, v. 5, p. 164–166, 1989.

FERLANTI, E. S. et al. Webblast 2.0: an integrated solution for organizing and analyzing sequence data. *Bioinformatics Application Note*, v. 15, p. 422–423, 1999.

FOUNDATION, T. A. S. *Apache*. November 2011. Disponível em: <http://www.apache.org-/>.

FRISHMAN, D. Functional and structural genomics using pedant. *Bioinformatics*, v. 17, p. 44–57, 2001.

IBM. *IBM Websphere*. November 2011. Disponível em: <www.ibm.com/software/br-/websphere/>.

LAGNEL, J.; TSIGENOPOULOS, C. S.; LLIOPOULOS, L. Noblast and jamblast: New options for blast and a java application manager for blast results. *Bioinformatics (2009).*, v. 25, p. 824–826, 2009.

LUDWIG, W. et al. Arb: a software environment for sequence data. *Nucleic Acids Research (2004).*, v. 32, p. 1363–1371, 2004.

MARSHALL, O. J. Perlprimer: cross-platform, graphical primer design for standard, bisulphite and real-time pcr. *Bioinformatics*, v. 20, p. 2471–2472, 2004.

MOON, J.; LEFKOWITZ, E. *XS BLAST*. November 2011. Disponível em: <http://www-.poxvirus.org/blast_xml_start.asp>.

NG, E. Y. K.; PANG, M. P. Comparison of nucleotide dna alignment search programmes. *Int. J. Medical Engineering and Informatics (2010).*, v. 2, 2010.

OLSEN, G. J. et al. Fastdnaml: a tool for construction of phylogenetic trees of dna sequences using maximum likelihood. *Computing Applied to Bioscience*, v. 10, p. 41–48, 1994.

ORACLE. *Java EJB*. November 2011. Disponível em: <http://www.oracle.com/technetwork-/java/javaee/ejb/index.html>.

PEARSON, W. R.; LIPMAN, D. J. The embl nucleotide sequence database. *Proceedings of the National Academy of Sciences (1988).*, v. 85, p. 2444–2448, 1988.

PORCINO, D.; HIRT, W. Ultra-wideband radio technology: Potential and challenges ahead. *IEEE Communications Magazine*, v. 41, p. 66–74, 2003.

RIDLEY, M. *Genome*. 1st. ed. [S.l.]: Harper Perennial - NY, 2006.

SCHRETTER, C.; MILINKOVITCH, M. C. Oligofaktory: a visual tool for interactive oligonucleotide design. *Bioinformatics*, v. 22, p. 115–116, 2005.

SCHULER, G. D. et al. Entrez: Molecular biology database and retrieval system. *Methods in Enzymology - Elsevier*, v. 266, p. 141–162, 1996.

SCHULTZ, J. et al. Smart: a web-based tool for the study of genetically mobile domains. *Nucleic Acids Research*, v. 28, p. 231–234, 2000.

SELLERS, P. H. Sellers. *Bull. Math. Biol.*, v. 46, p. 501–514, 1978.

SOH, D. et al. Enabling more sophisticated gene expression analysis for understanding diseases and optimizing treatments. *ACM SIGKDD Explorations*, v. 9, p. 3–14, 2007.

STAMATAKIS, A. P. et al. Accelerating parallel maximum likelihood-based phylogenetic tree calculations using subtree equality vectors. *Proc. Supercomputing Conference*, 2002.

STOESSER, G. et al. The embl nucleotide sequence database. *Nucleic Acids Research (2002).*, v. 30, p. 21–26, 2002.

STRIMMER, K.; HAESELER, A. von. Quartett puzzling: a quartett maximum likelihood method for reconstructing tree topologies. *Molecular Biology Evolution*, v. 13, p. 964–969, 1996.

TATENO, Y. et al. Dna data bank of japan (ddbj) in collaboration with mass sequencing teams. *Nucleic Acids Research (2000).*, v. 28, 2000.

TEAM, A. *Apache Tomcat*. November 2011. Disponível em: <http://tomcat.apache.org/>.

TEAM, M. D. *MySQL*. November 2011. Disponível em: <http://dev.mysql.com/>.

TEAM, O. D. *OpenMotif*. November 2011. Disponível em: <http://www.openmotif.org/>.

TEAM, P. D. *PBS Works*. November 2011. Disponível em: <http://www.pbsworks.com/>.

TEAM, P. D. *PHP*. November 2011. Disponível em: <http://www.php.net/>.

TEAM, S. D. *SuSE Linux*. November 2011. Disponível em: <http://www.suse.com/>.

ULLMAN, J. E. H. e J. D. *In Introduction to Automata Theory, Languages, and Computation*. [S.l.]: Addison-Wesley, 1979.

WATERMAN, R. F. S. e M. S. Smith e waterman. *Advan. Appl. Math*, v. 2, p. 482–489, 1981.

WECKX, S. et al. Snpbox: a modular software package for large-scale primer design. *Bioinformatics*, v. 21, p. 385–387, 2004.

WUNSCH, S. B. N. e C. D. Needleman e wunsch. *J. Mol. Biol.*, v. 48, p. 443–453, 1970.

WUYTS, J. et al. The european database on small subunit ribosomal rna. *Nucleic Acids Research*, v. 30, p. 183–185, 2002.

APÊNDICE A – Artigo IADIS

O apêndice A apresenta o artigo intitulado **VNBlast - Netblast System Management** que foi apresentado no dia 07/11/2011 no Congresso Internacional IADIS - Applyed Computing no prédio da Universidade Federal do Rio de Janeiro - UNIRIO - Rio de Janeiro - RJ.

VNBLAST: A NETBLAST MANAGEMENT SYSTEM

Marcio Rodrigo Santos
Universidade Estadual de Londrina
marcio@uel.br

Wesley Attrot
Universidade Estadual de Londrina
wesley@uel.br

Emerson José Venancio
Universidade Estadual de Londrina
emersonj@uel.br

ABSTRACT

GenBank is a public database of nucleotide sequences built by the National Center for Biotechnology Information (NCBI) that provide mechanisms for accessing and processing stored information. A way to access information on GenBank is BLAST (Basic Local Alignment Search Tool) a program to search for local similarity between sequences. GenBank information can be accessed through NCBI website or locally. NCBI BLAST website is an easy way to find sequences, but imposes some limitations in the parameterization of the query, and batch searches in large-scale are not available. In order to fill this gap, this paper will introduce the VNBlast tool. VNBlast is a user-friendly web application based on NCBI NetBlast, which offers a substantial number of parameters for search and alignment of sequences directly on GenBank, avoiding the need of manual download of datasets and always presenting as result updated information.

KEYWORDS

VNBlast; BLAST; GUI; Bioinformatics

1. INTRODUCTION

Biomolecular research is very important to better understanding of life and diseases (Soh, 2007). To improve biomolecular researches, a great effort has been done to describe genome from organisms, like human genome project (Ridley, 2000; Ng, 2010). The genome sequencing project and other sequencing strategies have generated a large number of nucleotide sequences. The best way to make this kind of information available for researchers all over the world is through public databases.

GenBank is a public database of nucleotide sequences of more than 380,000 organisms that was built by the National Center for Biotechnology Information (NCBI) from the submission of nucleotide sequences obtained by researchers (Benson, 2011).

But as important as the storage of information in databases is the process of finding specific information on the large amount of information stored. Large databases such as GenBank (Benson, 2011) provide mechanisms for accessing and processing stored data. GenBank information can be accessed through NCBI website or locally.

There are two main ways to access information on GenBank: ENTREZ (Schuler, 1996) and BLAST (Altschul, 1997). ENTREZ is an access system that supports search for text identifiers and annotations of sequences, and allows the obtaining of information in different formats. The BLAST - Basic Local Alignment Search Tool is a program to search for local similarity between sequences and it has been widely used to make inferences about the function of a given sequence and its possible phylogenetic relationships. Another important application of BLAST is the identification of conserved regions in sequences, with the objective of designing specific probes for reactions of molecular detection (Marshall OJ., 2004; Weckx et al, 2005; Schretter, 2006, Bally et al, 2010).

However, due to the exponential growth in the number of sequences available in GenBank, the search for information often demands a prohibitive time. In addition, the web interface of the BLAST program, available on the NCBI website, imposes some limitations in the parameterization of the query, and batch searches in large-scale are not available.

On the other hand, NCBI provides some tools available in text mode, with high processing capacity like the BLAST programs family (Blastn, Blastp, Blastx, Tblastn, Tblastx, Mega BLAST and Psi BLAST). Most of these tools were developed to perform searches and alignments of sequences against the GenBank. Despite its capacity and performance, sometimes, the command line can become extensive depending on the number of parameters necessary for its execution, increasing the chance of typos and the consequent failure on the process.

In order to increase the user interactivity and friendliness in the aligning process, this paper will introduce VNBlast, a graphical web version of NetBlast command line. For this purpose, this work will be segmented as follows: Section 2 will present some related work, Section 3 presents the VNBlast tool, Section 4 presents experimental results of the work, Section 5 presents the conclusion and future work and finally, section 6 presents the references.

2. RELATED WORK

Alkahest NuclearBLAST (Diener et al., 2005) is a web GUI for management and analysis system that was developed and packed as server application. This software provides one MySQL database to store queries. Through the database information from BLAST queries, using blastall tool is possible to make analysis and mining of results. This tool can also provide Gene Ontology terms to sequences and options to export information in spreadsheet format. The source of information used on NuclearBlast is the BLAST datasets, manually downloaded from BLAST site (NCBI). These downloaded datasets will feed the local database for searches. The use of a local database can present good performance, but it can also result in outdated information, if the database is not constantly updated.

Another example is the couple of tools NOBLAST and JAMBLAST (Lagnel, 2009). NOBLAST is an open source program that provides a tabular output format for various NCBI BLAST programs without any use of a parser, and provides E-value correction in case of use of segmented BLAST database. JAMBLAST using the NOBLAST output is a GUI that allows the user to manage, view and filter the BLAST hits using a number of selection criteria. JAMBLAST is written in JAVA, but is not a server application. It requires the installation of MySQL database and JAVA jre locally to work properly.

BlastQuest (Farmerie, 2005) relies on database technology and provides web-enabled query, analysis, and visualization facilities for genomics data. The interface with the Gene Ontology and the KEGG pathway databases foster the biological workflow. Following the same pattern of NuclearBlast, BlastQuest uses a local datasets, manually downloaded from Genbank.

In the same manner, it's possible to make reference to other projects like XS BLAST (Moon, 2011), Web BLAST (Ferlanti, 1999), OCGC (Cuticchia, 1999) and PEDANT (Frishman, 2001). The main focus of those projects is the post-processing, taking as input BLAST results. But in general ways, there are only few parameters for searching and alignment, and the great part of processing is done after the return of the results from BLAST.

3. VNBLAST

BLAST command line tools offer many other features of searching and alignment not used by these similar referenced tools that can make the search process more accurate or show the results in different ways. The pre-processing itself can offer great final results. Another version of BLAST is NetBlast or BLAST client, also included on recent BLAST+ version as "-remote" option. These versions of BLAST have the advantage of taking the information from GenBank through webservices, avoiding the need of manual download of datasets from NCBI. One important advantage of using these tools is that the retrieved information is always updated. In order to fill this gap, this paper introduces VNBlast as one interesting alternative to the NCBI BLAST website for searches and alignments of genetic sequences.

VNBlast (Visual NetBlast) is a web application that was developed in order to address the limitations existing in the current search and alignment engine available on NCBI for web access. This process occurred through the creation of a web GUI which allowed the use of the resources currently available only in local text mode application, but offering a new multi-platform user-friendly web interface.

The choice of programming tools was one important step in this project. The chosen language should be able to handle both web environments, and manage applications such as Netblast. For this purpose, Java was chosen to be the main language and JBoss (JBoss, 2011) for server application, once VNBlast was developed in layers. The chosen tool for development of GUI was Adobe Flex (Adobe Flex, 2011). Flex is a language for development of web GUI, Java compatible which have good resources for the design of friendly interfaces under RIA – Rich Internet Applications concept. Resources like forms, combo boxes, tooltips and grids are available on this tool. These chosen tools are free for use on web.

One of the objectives of VNBlast project was the reuse of one stable application in its binary form, reducing significantly the time of development. But for this, one important point to be checked is the capacity of management of external applications by the chosen programming language. The control of the exact moment of the initialization and completion of the console application is of fundamental importance in this case, once is the operational system who handles the status of processes. Java has java.lang.Runtime class, which exchange information with OS, allowing the view of process status.

In VNBlast, the process of searching and alignment of genetic sequences became simpler and easier for the user. The GUI has combo boxes for pre-defined parameters that avoid typos. The selection of the criteria file is also easier through the dialog box. All the fields on form have a tooltip presenting its description and an example of use. Unlike Netblast that saves the results to a file, VNBlast automatically display the results in a form field, allowing the subsequent download of the resulting file. Figure 1 shows the grayscale version of search window displaying the results of an alignment.

Figure 1. VNBlast Search Form in grey scale

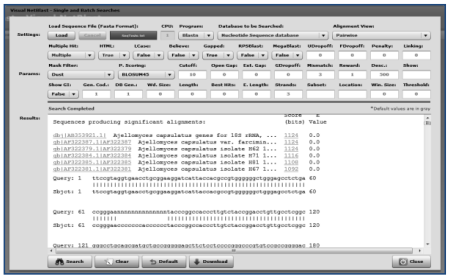


Figure 2. Outstanding non-default options

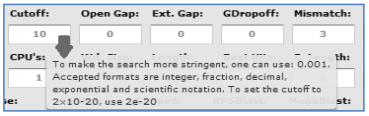
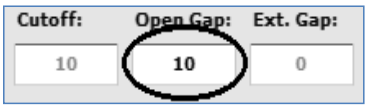


Figure 3. Informative Tooltips



In the results box, some items are displayed as hyperlinks. A mouse click on one of these items will lead to an information page showing additional information about this specific item.

NetBlast parameters are now implemented in a visual way, offering graphical resources as highlighting options such as bold fields for values defined different than BLAST default, as presented in Figure 2.Another useful graphic resource available in VNBlast is the tooltips on system fields activated on "mouse-over" event. On labels, it presents the meaning of parameter and on fields it displays the accepted format and samples of use. Figure 3 shows a tooltip of the Cutoff parameter field.

The selection process of the query sequences file is made now in graphical manner through dialog box. Fields with limited and pre-defined values were displayed in combo-boxes, avoiding typos and making the parameter definition much easier. Successive consultations with minor changes can be easily done without the necessarily download of resultant file, once the results are displayed on the form. In this way, the final result can be reached in less time. Some values of parameters imply in the settings of another parameter, and VNBlast is automatically able to manage these situations.

Using the NetBlast view options, VNBlast provides 11 options to view the results of alignments like Pairwise and variations, XML Blast Output, Tabular with and without comments, Query Anchored showing identities or not, Flat Query Anchored and so on. After the desired results been displayed on the Results field, it is possible to save the results in a file through download button option.

4. CONCLUSION AND FUTURE WORK

VNBlast offered an advantage over the NCBI BLAST search and align website, mainly due to the number of parameters available for search, filter and align the sequences against the GenBank database.

Another improvement is the capacity of perform batch searches, that are unavailable on the NCBI BLAST site. The only system requirement for accessing VNBlast is a web browser with Adobe Flash plug-in installed. When comparing to a command line applications, VNBlast offers not only a GUI, but a web environment system, able to deal with multiple simultaneous searches, creating a new virtual session for each call. When compared to related works, VNBlast have its focus on the pre-processing, presenting a bigger number of pre-processing options, making the searches directly in GenBank and taking always updated information. VNBlast still does not have post-processing options and its improvement is over similar NCBI BLAST searches, providing better options of filtering and alignment and availability of batch searches. VNBlast is available for access under the web address: http://vnblast.dc.uel.br:8080/genbank.

As future work, we will create (in progress) a mechanism for handling strings that will process the results of queries generated by VNBlast, concatenating the genetic sequences fragmented and subsequently converting them into one Netblast compatible input format. Once the conversion is complete, it will be possible to feed back the resulting sequence in a new query. Another improvement will be the inclusion of statistics graphics and reports giving the researcher a better view of the results.

REFERENCES

Ridley, M., 2000. Genome, Harper Collins, October 2000.

Ng, E.Y.K, Pang, M.P., 2010. *Comparison of nucleotide DNA alignment search programmes*, Int. J. Medical Engineering and Informatics, Vol. 2, No. 2, pp.163–176.

Benson, D.A. et al., 2011, *GenBank*. Nucleic Acids Res. 2011. Jan; 39(Database issue):D32-7

Soh, D. et al., 2007. *Enabling more sophisticated gene expression analysis for understanding diseases and optimizing treatments*, ACM SIGKDD Explorations Newsletter, Volume 9 Issue 1, June 2007.

Schuler, G.D. et al., 1996. *Entrez: molecular biology database and retrieval system*. Methods Enzymol. 266:141-62.

Altschul, S.F. et al., 1997. *Gapped B.L.A.S.T. and PSI-B.L.A.S.T.: a new generation of protein database search programs*. Nucleic Acids Res.; 25:3389–3402.

Marshall, O.J., 2004. *PerlPrimer: cross-platform, graphical primer design for standard, bisulphite and real-time PCR*. Bioinformatics. 2004 Oct 12;20(15):2471-2.

Weckx, S. et al., 2005. *SNPbox : a modular software package for large-scale primer design*. Bioinformatics. 2005 Feb 1;21(3):385-7.\

Schretter, C., Milinkovitch, M.C., 2006. *OligoFaktory: a visual tool for interactive oligonucleotide design*. Bioinformatics. 2006 Jan 1;22(1):115-6.

Bally, P. et al., 2010. *FONZIE: An optimized pipeline for minisatellite marker discovery and primer design from large sequence data sets.* BMC Res Notes. 2010 Nov 29;3:322.

BLAST, NCBI, USA, < http://blast.ncbi.nlm.nih.gov/Blast.cgi> Last viewed on 23 August 2011.

Diener, S. E. et al., 2005. *Alkahest NuclearBLAST: a user-friendly BLAST management and analysis system*. BMC Bioinformatics, 6, 147

Lagnel, J. et al., 2009. *NOBLAST and JAMBLAST: New Options for BLAST and a Java Application Manager for BLAST results*. Bioinformatics Vol. 25, No. 6. (29 January 2009), pp. 824-826. doi:10.1093.

Moon, J. and Lefkowitz, E., *XS BLAST (XML–SQL BLAST)*. Available from http://www.poxvirus.org/blast_xml_start.asp Last viewed on 29/08/2011.

Ferlanti, E. S. et al., 1999. *WebBLAST 2.0: an integrated solution for organizing and analyzing sequence data*. Bioinformatics 15 422–423.

Cuticchia, J. et al., 1999. *OCGC Blast*,. Available on http://www.ocgc.ca/ocgcblast.htm. Last viewed on 23 August 2011.

Frishman, D. et al., 2001. *Functional and structural genomics using PEDANT*. Bioinformatics 17 (1) 44–57.

Farmerie, W.G. et al., 2005. *Biological workflow with BlastQuest*. Data Knowl. Eng. 75-97 Elsevier.

JBoss Comunity, *JBoss*, http://www.jboss.org/overview.html. Last viewed on 23 August 2011

Adobe, *Flex Builder*, http://www.adobe.com/products/flex/. Last viewed on 23 August 2011

APÊNDICE B – Pôster CICPG

O apêndice B apresenta o pôster intitulado **Acesso, Busca e Alinhamento de Sequências Genéticas ao Banco de Dados Genbank** que foi apresentado no CICPG - Congresso de Iniciação Científica e Pós-Graduação - Sul Brasil em setembro de 2010 em Florianópolis - SC.

CONGRESSO DE INICIAÇÃO CIENTÍFICA E PÓS-GRADUAÇÃO - SUL BRASIL

ACESSO, BUSCA E ALINHAMENTO DE SEQÜÊNCIAS GENÉTICAS AO BANCO DE DADOS GENBANK

Marcio Rodrigo Santos, Wesley Attrot, Alan Salvanny Felinto, Jacques Duílio Brancher
Departamento de Computação - Centro de Ciências Exatas, Universidade Estadual de Londrina (UEL)
Palavras-chave: Genbank, Alinhamento genético, Netblast.

INTRODUÇÃO

A bioinformática tem desempenhado um importante papel no auxílio à medicina no que tange a preservação da vida, através da descoberta de novas vacinas e do tratamento de doenças de origem genética e não genética, através de técnicas biomoleculares.
Quando uma nova doença surge, efetuar o mapeamento genético do organismo originador é fundamental para a criação do medicamento. Desta forma, reduzir o intervalo de tempo entre a descoberta da doença e o desenvolvimento do agente de combate correto também é muito importante. Tanto neste quanto em outros casos, uma das principais formas de detecção e comparação entre organismos vivos é através do seqüenciamento e alinhamento genéticos.

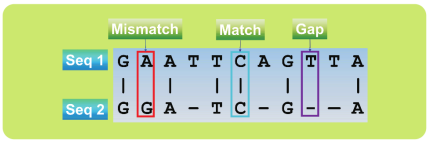


Fig. 1 – Exemplo de alinhamento entre duas seqüências

Fig. 2 – Alinhamento Global x Alinhamento Local

Obter as seqüências genéticas envolvidas é outro passo importante no processo de pesquisa. Com este objetivo existem alguns grandes bancos de dados de informações genéticas. O Genbank é um dos maiores bancos de dados de informações genéticas do mundo. Seus dados estão disponíveis ao público em geral para acesso, inclusão e consulta.
Dentre as formas disponíveis para acesso a estas informações, estão o método web, download de arquivos de dados e web-services. A versão web oferece a facilidade do acesso através de web-browsers, porém possui algumas restrições de funcionamento.

NCBI GenBank

OBJETIVOS

O presente projeto tem como objetivo oferecer uma ferramenta que reúna algumas das melhores características dos diversos métodos de acesso, facilitando o acesso e consulta ao Genbank

METODOLOGIA

Decorrente desta necessidade foi criada uma ferramenta chamada G.A.S. – Genetic Alignment Search. Trata-se de uma ferramenta web acessível através de navegadores de internet. Esta ferramenta agrega algumas das melhores características dos métodos de acesso a informações do Genbank atualmente disponíveis. Dentre elas destacamos o razoável número de parâmetros de consulta e filtragem disponíveis e a forma de acesso através da web.

RESULTADOS E DISCUSSÃO

No projeto G.A.S. foram utilizadas ferramentas com tecnologia Rich Internet Application (R.I.A.), que permitem a inclusão de recursos desktop, como tooltips descritivos nos campos.

Fig. 3 – Fluxo de informações no projeto G.A.S.

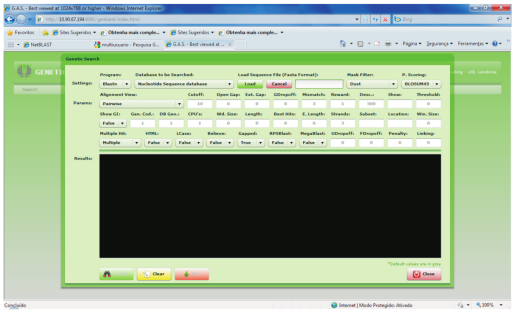


Fig. 4 – Um panorama geral da ferramenta

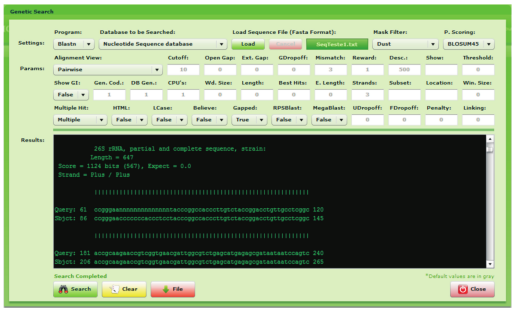


Fig. 5 – O resultado de uma consulta efetuada.

CONCLUSÕES

Através do projeto G.A.S., foi possível unir a praticidade que a web oferece com o poder que os mecanismos de consulta através de WebService disponibilizam, apresentando um bom desempenho no tempo de resposta das consultas efetuada.

REFERÊNCIAS BIBLIOGRAFICAS

1. Altschul, S. F., et al. Basic Local Alignment Search Tool. Journal of Molecular Biology 215, 403–410 (1990) doi:10.1016/S0022-2836(05)80360-2).
2. O'REILLY, Tim. What Is Web 2.0 - Design Patterns and Business Models for the Next Generation of Software. O'Reilly Publishing, 2005

APÊNDICE C – Formato FASTA

Em bioinformática, o formato FASTA (PEARSON; LIPMAN, 1988) é um padrão baseado em texto para representar tanto sequências de nucleotídeos quanto de peptídeos, no qual os nucleotídeos ou aminoácidos são representados através de códigos de uma única letra. Este formato se originou do software FASTA (PEARSON; LIPMAN, 1988) e se tornou um padrão de formatação na área de bioinformática. Uma sequência em formato FASTA é iniciada por uma descrição de uma única linha, sucedida por linhas que contém a descrição das bases nitrogenadas em sequência. A linha de descrição (*defline*) é separada dos dados da sequência por um símbolo de **maior que** ($>$) no início. É recomendável que todas as linhas do texto sejam menores do que 80 caracteres. A Figura C apresenta um exemplo de sequência em formato FASTA do fungo Ajellomyces capsulatus var. farciminosus 18S.

Linhas em branco não são permitidas no meio de uma entrada de sequência no formato FASTA. As sequências devem ser representadas no padrão IUB/IUPAC para aminoácidos e ácidos nucléicos, com as seguintes exceções: letras minúsculas são aceitas e são mapeadas como letras maiúsculas; um único hífen ou traço pode ser usado para representar um espaço de comprimento indeterminado, e em sequências de aminoácidos, U e * são letras aceitáveis.

```
>gi|12584196|gb|AF322387.1| Ajellomyces capsulatus var. farciminosus 18S
TTCCGTAGGTGAACCTGCGGAAGGATCATTACCACGCCGTGGGGGGCTGGGAGCCTCTGACCGGGAACCCCC
CCACCCCCCTACCCGGCCACCCTTGTCTACCGGACCTGTTGCCTCGGCGGGCCTGCAGCGATGCTGCCGGGG
GAGCTTCTCCTCCCCGGGCCCGTGTCCGCCGGGGACACCGCAAGAACCGTCGGTGAACGATTGGCGTCTGAG
CATGAGAGCGATAATAATCCAGTCAAAACTTTCAACAACGGATCTCTTGGTTCCGACATCGATGAAGAACGC
AGCGAAATGCGATAAGTAATGTGAATTGCAGAATTCCGTGAATCATCGAATCTTTGAACGCACATTGCGCCC
CCTGGTATTCCGGGGGGCATGCCTGTCCGAGCGTCATTGCAACCCTCAAGCGCGGCTTGTGTGTTGGGCCGT
CGTCCCCCCTCGACCGGCGGGACGTGCCCGAAATGCAGTGGCGGTGTCGAGTTCCGGTGCCCGAGCGTATGG
GGCTTTGCCACCCGCTCTGGAGGCCCGGCCGGCTCCGGCCCACCATGTCAACCCCCCTCTCACACCAGGTTG
ACCTCGGATCAGGTAGGGATACCCGCTGAACTT
```

Figura C.1: Exemplo de sequência FASTA do fungo Ajellomyces capsulatus var. farciminosus 18S