UNIVERSIDADE DE CAXIAS DO SUL

DANIEL ORLANDI MINCATO

INTEGRAÇÃO DE DADOS E MIGRAÇÃO WEB DA FERRAMENTA NET2HOMOLOGY

CAXIAS DO SUL

2014

DANIEL ORLANDI MINCATO

INTEGRAÇÃO DE DADOS E MIGRAÇÃO WEB DA FERRAMENTA NET2HOMOLOGY

Trabalho de Conclusão de Curso apresentado como requisito parcial para obtenção do grau de Bacharel pela Universidade de Caxias do Sul, na área da Ciência da Computação.

Orientador Prof. Dr. Daniel Luis Notari.

CAXIAS DO SUL

2014

AGRADECIMENTOS

Agradeço aos meus familiares, meu pai Remo Mincato, minha mãe Ivete Orlandi Mincato e minha irmã Claudia Orlandi Mincato pelo apoio e força dados à mim para que eu alcançasse aos meus objetivos.

Ao meu orientador, o professor Daniel Luis Notari, pela dedicação e paciência prestados durante o desenvolvimento desse trabalho, com seus conselhos e opiniões que tornaram a concretização dele possível.

Aos amigos Janaína Graciele Steffens e Mateus Izoton pelo convívio sempre alegre e divertido que aliviaram os momentos de estresse vividos durante o período de realização da minha graduação.

A todos os professores com os quais tive a oportunidade de aprender, não somente durante o período da minha graduação, mas de toda a vida e que, sem o quais, não teria conseguido concluir a minha graduação.

RESUMO

A bioinformática tem disponibilizado cada vez mais ferramentas computacionais capazes de armazenar, recuperar, analisar e comparar grandes volumes de dados biológicos, sendo indispensáveis para a criação de *workflows* científicos. Os *workflows* científicos dão suporte às pesquisas, auxiliando na preparação e execução das tarefas cotidianas dos pesquisadores, como a execução de um experimento ou a análise de um conjunto de dados. A aplicação Net2Homology foi desenvolvida como *workflow* científico que compara duas redes de interação proteína-proteína, de dois organismos diferentes, verificando as estruturas e funções conservadas no processo evolutivo. Um problema dessa aplicação está na integração de dados entre os bancos de dados do NCBI e o banco de dados STRING. Isto acontece no momento da extração da proteína do relatório do BLAST, quando também busca a referência dessa proteína no STRING. Consequentemente, algumas redes de interação são trazidas de forma errônea, impossibilitando a comparação entre elas. Neste trabalho foi proposta e realizada uma integração de dados utilizando um banco de dados local que armazena referências diretas entre os dados dos bancos de dados citados, ao mesmo tempo que a aplicação Net2Homology foi migrada para a plataforma *web*, utilizando o *framework* JSF, a biblioteca de componentes PrimeFaces e o servidor *web* Apache Tomcat.

Palavras-chave: Bioinformática. *Workflow* científico. Integração de dados.

ABSTRACT

The experiments made using bioinformatics tools allows users to store, retrieve, analyze and compare large amounts of biological data. Scientific workflows serves as a guide to execute bioinformatics experiments. The scientific workflows gives support to researches, assisting in the preparation and execution of daily tasks of researchers, like an experiment execution or analyze of a data set. The Net2Homology application was built as a scientific workflow that compares two protein-protein interaction networks, from two different organisms, checking the structures and functions conserved in the evolution process. A problem of this application is centered when the protein is extracted from BLAST report to look for a STRING's reference match. Consequently, some interactions are brought to the application with error, which made it impossible to compare. In this paper, it was proposed and held a data integration using a local database that stores direct references between data of cited databases at the same time that the Net2Homology application was migrated to web platform.

Key words: Bioinformatics. Scientific Workflow. Data Integration.

LISTA DE ILUSTRAÇÕES

Figura 1 - Exemplo de utilização da API E-utilities com URL. 18
Figura 2 - Exemplo de Utilização da API E-utilities com SOAP. 18
Figura 3 - Exemplo de utilização da STRING API. 19
Figura 4 - Exemplo de resultado do algoritmo BLASTp. 21
Figura 5 - Exemplo de resultado do algoritmo PSI-BLAST no formato XML. 22
Figura 6 - Fluxo de execução do workflow Net2Homology 26
Figura 7 - Tela inicial da aplicação Net2Homology. 29
Figura 8 - Tela de configuração da aplicação Net2Homology. 30
Figura 9 - Tela de seleção de proteínas e organismo origem. 31
Figura 10 - Tela de configuração do BLAST. 32
Figura 11 - Tela de seleção de proteínas e consulta de redes de interação proteína-proteína. .33
Figura 12 - Esboço da tela de seleção de proteínas e organismo origem. 35
Figura 13 - Esboço da tela de configuração do BLAST. 36
Figura 14 - Esboço da tela de consulta ao banco de dados STRING. 37
Figura 15 - Esboço da tela de visualização do resultado da consulta ao STRING. 38
Figura 16 - Fluxo do algoritmo de busca do identificador no banco de dados STRING. 39
Figura 17 - Diagrama entidade-relacionamento da aplicação Net2Homology. 40
Figura 18 - Arquitetura MVC na aplicação Net2Homology. 41
Figura 19 - Estrutura de *packages* e pastas do projeto Net2HomologyWeb. 44
Figura 20 – Estudo de Caso na tela de seleção de organismo origem e proteínas. 47
Figura 21 - Estudo de caso na tela de configuração do BLAST 48
Figura 22 – Estudo de caso na tela de resultado do BLAST e consulta ao STRING. 49
Figura 23 - Estudo de caso na tela de resultado da consulta ao STRING. 50
Figura 24 - Comparação dos resultados obtidos das execuções *desktop* e *web*.....51
Figura 25 - Verificação das referências das proteínas no banco de dados local. 52
Figura 26 - Tela de seleção de organismo origem e proteínas na plataforma *web*. 53
Figura 27 - Tela de seleção do organismo alvo e configuração do BLAST. 54
Figura 28 - Tela de resultado do BLAST e consulta ao STRING. 56
Figura 29 - Tela de visualização das redes PPI. 57
Figura 30 - Botão para reinício do *workflow*. 58

LISTA DE TABELAS

Tabela 1 - Comparação entre os bancos de dados biológicos. 24
Tabela 2 - Comparação entre as ferramentas. 28
Tabela 3 - Comparação de funcionalidades entre as versões da ferramenta Net2Homology. 46

LISTA DE SIGLAS

API	*Application Programming Interface*	Interface de Programação de Aplicativo
BLAST	*Basic Local Alignment Search Tool*	Ferramenta Básica de Busca de Alinhamento Local
DDBJ	*DNA Databank of Japan*	Banco de Dados de DNA do Japão
EBI	*European Bioinformatics Institute*	Instituto Europeu de Bioinformática
EMBL	*European Molecular Biology Laboratory*	Laboratório Europeu de Biologia Molecular
EST	*Expressed Sequence Tag*	Tag de Expressão de Sequência
FTP	*File Transfer Protocol*	Protocolo de Transferência de Arquivo
GSS	*Genome Survey Sequence*	Análise de Sequência de Genoma
HTTP	*Hyper Text Transfer Protocol*	Protocolo de Transferência de Hipertexto
INSDC	*International Nucleotide Sequence Database Collaboration*	Colaboração Internacional de Bancos de Dados de Sequências de Nucleotídeos
IP	*Internet Protocol*	Protocolo de Internet
JAR	*Java Archive*	Arquivo Java
JDBC	*Java Database Connectivity*	Conexão com Bancos de Dados em Java
JDK	*Java Development Kit*	Estojo para Desenvolvimento Java
JRE	*Java Runtime Environment*	Mecanismo de Execução Java
JSF	*Java Server Faces*	Servidor de Telas em Java
JSP	*Java Server Pages*	Servidor de Páginas em Java
JVM	*Java Virtual Machine*	Máquina Virtual Java
MVC	*Model-View-Controller*	Modelo-Visão-Controlador
NCBI	*National Center of Biotechnology Information*	Centro Nacional de Informações em Biotecnologia
PIR	*Protein Information Resource*	Recurso de Informações de Proteínas
PPI	*Protein-Protein Interaction*	Interação Proteína-Proteína
PSI-BLAST	*Position Specific Iterated – Basic Local Alignment Search Tool*	Ferramenta Básica de Busca de Alinhamento Local Iterada em Posição Específica
SEMEDA	*Semantic Meta Database*	Banco de Dados de Semântica de Meta Dados
SOAP	*Simple Object Access Protocol*	Protocolo Simples de Acesso à Objeto
SQL	*Structured Query Language*	Linguagem de Pesquisa Estruturada

STRING	*Search Tool for the Retrieval of Interacting Genes/Proteins*	Ferramenta de Pesquisa para Recuperação de Interações de Genes/Proteínas
TSV	*Tab-Separated Values*	Valores Separados por Tabulação
URL	*Uniform Resource Locator*	Localizador Uniforme de Recursos
XML	*Extensible Markup Language*	Linguagem de Marcação Extensível

SUMÁRIO

1 INTRODUÇÃO 12

2 BANCOS DE DADOS BIOLÓGICOS 15

2.1 GENBANK 15

2.2 SWISS-PROT 15

2.3 STRING 16

2.4 ACESSO À BANCOS DE DADOS BIOLÓGICOS 17

2.4.1 Entrez 17

2.4.2 STRING API 19

2.5 ANÁLISE DE HOMOLOGIA 19

2.5.1 BLASTp 20

2.5.2 PSI-BLAST 21

2.6 INTEGRAÇÃO DE BANCOS DE DADOS E SISTEMAS 23

2.7 CONSIDERAÇÕES FINAIS 23

3 WORKFLOW CIENTÍFICO 25

3.1 NET2HOMOLOGY 25

3.2 FERRAMENTAS SEMELHANTES 26

3.2.1 BioExtract 27

3.2.2 SEMEDA 27

3.3 CONSIDERAÇÕES FINAIS 28

4 PROPOSTA DE SOLUÇÃO 29

4.1 A APLICAÇÃO NET2HOMOLOGY ATUAL 29

4.1.1 Configuração da Ferramenta 30

4.1.2 Utilização do *Workflow* 30

4.2 PROBLEMA ATUAL 34

4.3 A NOVA APLICAÇÃO 34

4.3.1 Seleção de Organismo Origem e Proteínas 34

4.3.2 Configuração do BLAST 35

4.3.3 Consulta ao STRING 37

4.3.4 Visualização dos Resultados da consulta ao STRING 38

4.3.5 Integração NCBI x STRING 39

4.3.6 Implementação 40
4.3.6.1 Arquitetura Física 40
4.3.6.2 Arquitetura Lógica 41
4.3.7 Testes 42
4.4 CONSIDERAÇÕES FINAIS 42

5 DESENVOLVIMENTO 43
5.1 PREPARAÇÃO DO AMBIENTE 43
5.2 O PROJETO 43
5.3 CODIFICAÇÃO 44
5.4 TESTES E HOSPEDAGEM 45
5.5 CONSIDERAÇÕES FINAIS 46

6 ESTUDO DE CASO 47
6.1 TELA DE SELEÇÃO DE ORGANISMO ORIGEM E PROTEÍNAS 47
6.2 TELA DE CONFIGURAÇÃO DO BLAST 48
6.3 TELA DE RESULTADO DO BLAST E CONSULTA AO STRING 48
6.4 TELA DE RESULTADO DA CONSULTA AO STRING 50
6.5 INTEGRAÇÃO NCBI X STRING 51
6.6 FUNCIONALIDADES DA APLICAÇÃO 53
6.6.1 Tela de Seleção de Organismo Origem e Proteínas 53
6.6.2 Tela de Seleção do Organismo Alvo e Configuração do BLAST 54
6.6.3 Tela de Resultados do BLAST e Consulta ao STRING 55
6.6.4 Tela de Visualização das Redes PPI 57
6.6.5 Reiniciar o *Workflow* 58

7 CONSIDERAÇÕES FINAIS 59

8 REFERÊNCIAS 61

1 INTRODUÇÃO

Com o surgimento dos grandes centros de sequenciamento, uma quantidade de dados cada vez maior está sendo gerada e precisa ser processada e analisada. Para suprir essa demanda, a bioinformática provê ferramentas computacionais, desde o armazenamento e recuperação desses grandes volumes de dados, até a utilização de métodos quantitativos e empíricos de análise para processá-los e compará-los (GIBAS; JAMBECK, 2001). Deste modo, apresentando os resultados obtidos dessas análises aos pesquisadores, agilizando o desenvolvimento de seus trabalhos.

Ao realizarem uma pesquisa no campo da biologia molecular, os biólogos utilizam as ferramentas providas pela bioinformática, como uma sequência de passos para a realização de experimentos. Essa sequência de passos, quando organizada em um fluxo de trabalho, transforma-se em um *workflow* científico (SILVA; CAVALCANTI; DÁVILA, 2006) que pode ser construído como uma ferramenta computacional e, neste caso, cada passo do *workflow* pode ser a invocação de um programa ou a chamada de um serviço *web* (GIL et al., 2007).

Os *workflows* e ferramentas que analisam as proteínas que fazem parte de uma função celular têm a capacidade de processá-las juntamente com as interações existentes entre elas. Essas interações podem ser representadas na forma de grafos de redes, sendo denominadas redes de interação proteína-proteína. Nessas redes, os nodos são as proteínas e as conexões são as relações existentes entre elas (LU et al., 2004). Essas relações indicam ligações químicas entre as proteínas e podem caracterizar relações físicas ou funcionais existentes entre os pares de proteínas conectadas.

A aplicação Net2Homology foi desenvolvida no formato de *workflow* científico. O objetivo dessa ferramenta é efetuar a comparação de duas redes de interação proteína-proteína, de dois organismos diferentes, verificando as estruturas e funcionalidades conservadas durante o processo evolutivo (NOTARI, 2012). Nessa ferramenta são realizadas consultas nos bancos de dados de proteínas do *National Center of Biotechnology Information* (NCBI) e bancos integrados diretamente à eles como o Swiss-Prot, *DNA DataBank of Japan* (DDBJ), entre outros, e no banco de dados de interações entre proteínas STRING. Essa aplicação utiliza também a ferramenta de acesso Entrez, para acessar o NCBI, e a STRING API para acessar o STRING, ambas disponibilizadas pelos bancos de dados para acessar e analisar seus dados. Também utiliza a ferramenta NetworkBlast para realizar a comparação das redes de interação proteína-proteína, apresentando os resultados ao final de algumas etapas de seu fluxo.

Um problema encontrado na aplicação Net2Homology é uma falha na integração de dados entre os bancos de dados que acessa, quando está preparando a consulta das redes de interação proteína-proteína, ocasionando que não se consiga consultar todas as redes de interação por falha nessa preparação de dados. Outro problema é o fato de necessariamente precisar configurar um servidor *proxy* e uma porta de conexão para conseguir acessar as ferramentas que utiliza. Essa etapa de configuração, além de exigir conhecimento computacional para fazê-la, pode passar despercebida ao usuário, pois não há alertas sobre essa necessidade.

Com base nisso, o objetivo desse trabalho é aprimorar a aplicação Net2Homology, resolvendo os problemas de integração entre os bancos de dados do NCBI e o banco de dados STRING, e de configuração da ferramenta ao mesmo tempo que se disponibiliza ela na *web*, sendo acessada através do navegador, não necessitando mais efetuar o *download* e executá-la localmente. Para atingir esses objetivos foi proposta uma solução para a integração de dados utilizando um banco de dados local que armazena referências diretas entre os bancos de dados e, para disponibilizá-la na *web*, utilizando o *framework Java Server Faces* (JSF), a biblioteca de componentes PrimeFaces e servidor *web* Apache Tomcat. Essa proposta foi desenvolvida, criando a nova versão do *workflow* Net2Homology na plataforma *web* com o algoritmo de integração de dados aprimorado.

Este trabalho está organizado em três grandes seções, sendo a primeira parte conceitual, necessária para compreender as ferramentas que estão disponibilizadas e são utilizadas na aplicação Net2Homology. Essa parte conceitual está dividida nos capítulos 2 e 3 deste trabalho. O capítulo 2 trata de bancos de dados biológicos, métodos de acesso aos seus dados e ferramentas integradas a eles. Também aborda conceitos referentes à integração de bancos de dados e sistemas. O capítulo 3 deste trabalho aborda os conceitos de *workflow* científico, apresentando o *workflow* da aplicação Net2Homology e explicando outras ferramentas que tratam de problemas similares aos tratados no Net2Homology, como integração e grandes volumes de dados.

A segunda parte desse trabalho aborda a proposta de solução, sendo totalmente detalhada no capítulo 4. Esse capítulo inicialmente apresenta a aplicação Net2Homology atual, apresentando suas telas e detalhando suas funcionalidades, no que tange o escopo desse trabalho. Após é feito um detalhamento sobre os problemas da ferramenta. Por fim, é realizada a análise da nova aplicação, apresentando os requisitos e detalhando as funcionalidades que serão tratados, assim como apresenta os esboços de tela gerados e arquiteturas que serão utilizadas com base na análise efetuada.

A terceira parte desse trabalho trata sobre o desenvolvimento e análise dos resultados obtidos com a nova aplicação, sendo detalhada nos capítulos 5 e 6. O capítulo 5 descreve as etapas do desenvolvimento como a preparação do ambiente, o projeto criado, como a codificação aconteceu, como os testes foram realizados e como a hospedagem da aplicação na *web* foi efetuada. O capítulo 6 apresenta um estudo de caso, demonstrando cada uma das telas da nova aplicação, demonstradas com a utilização de dados reais, e comparando os resultados obtidos nessa nova versão com os dados da versão anterior da ferramenta. Nesse capítulo também é feito um detalhamento a respeito de todas as funcionalidades disponibilizadas, tela por tela, na aplicação Net2Homology na plataforma *web*.

2 BANCOS DE DADOS BIOLÓGICOS

Para o entendimento do que é um banco de dados biológico, torna-se necessária a compreensão de dois conceitos: bancos de dados e dados biológicos. Banco de dados é um repositório de informações que representam aspectos do mundo real, possuem coerência lógica entre si e inferem significado e interesse para um grupo de usuários (ELMASRI e NAVATHE, 2011). Dados biológicos são as unidades básicas de informação que representam fenômenos concretos ou abstratos associados a um contexto (PUGA, FRANÇA e GOYA, 2014), sendo que este contexto, neste caso, está contido na área da biologia. Com base nisso, bancos de dados biológicos são repositórios de informações que expressam significados dentro do contexto da biologia. Nas seções seguintes serão apresentados três bancos de dados biológicos disponíveis publicamente, algumas formas de acessos a esses bancos, ferramentas para análise de homologia disponibilizadas por eles e conceitos gerais sobre integração de dados e sistemas.

2.1 GENBANK

O GenBank é um banco de dados biológico, desenvolvido e distribuído publicamente pelo NCBI, que armazena sequências de nucleotídeos, sequências de aminoácidos, entre outras informações, de forma abrangente, apoiando a anotação bibliográfica e biológica (BENSON et al., 2014). A atualização das informações do GenBank ocorre, de forma sequencial, pela publicação de trabalhos dos pesquisadores ou, de forma massiva, pelo envio em larga escala de dados oriundos da *Expressed Sequence Tag* (EST), *Genome Survey Sequence* (GSS) e de dados originados de centros de sequenciamento (BENSON et al., 2014).

O Genbank atua com o *DNA DataBank of Japan* (DDBJ) e com o Banco de Dados de Sequências de Nucleotídeos do *European Molecular Biology Laboratory* do *European Bioinformatics Insitute* (EMBL-EBI) em uma colaboração internacional denominada *International Nucleotide Sequence Database Colaboration* (INSDC) que tem por objetivo prover dados de sequências e de informações associadas, uniformes e de forma abrangente (KARSCH-MIZRACHI; NAKAMURA; COCHRANE, 2012). As informações do GenBank podem ser acessadas através da *internet*, por meio de transferência de arquivos pelo protocolo FTP ou outros serviços *web* de recuperação e análise de sequências (BENSON et al., 2014).

2.2 SWISS-PROT

O Swiss-Prot é um banco de dados de sequências e conhecimento sobre proteínas, valorizado por suas anotações de alta qualidade, uso de nomenclatura padronizada, *links* diretos

com outros bancos de dados especializados e pela redundância mínima, formatando seus dados, sempre que possível, conforme a padronização proposta pelo Banco de Dados de Sequências de Nucleotídeos do EMBL (BOECKMANN et al., 2003). Ele possui referência direta aos dados dos bancos de dados de nucleotídeos do EMBL/DDBJ/GenBank, bancos de dados de estruturas 2D e 3D de proteínas, bancos de dados de caracterização de famílias e domínios de proteínas, bancos de dados de modificação pós-translacional, coleções de dados específicos de espécies, bancos de dados de mutações e bancos de dados de doenças (BOECKMANN et al., 2003).

O Swiss-Prot atua com *Protein Information Resource* (PIR) e o EBI de um consórcio internacional denominado UniProt onde compartilham seus bancos de dados e fornecem ferramentas de acesso para recuperação das informações (LESK, 2008). Estão relacionados mais proximamente com o SWISS-PROT os bancos de dados ENZYME DB que contém informações como nome recomendado, nomes alternativos, atividades catalíticas, entre outas informações das proteínas, e o PROSITE que possui informações de resíduos comuns entre grupos de proteínas, indicando relações que não são detectáveis pela comparação de sequências de aminoácidos (LESK, 2008).

2.3 STRING

O STRING é um banco de dados que tem por objetivo montar, avaliar e disseminar informações sobre associações proteína-proteína (FRANCESCHINI et al., 2013). As associações entre proteínas, conhecidas ou estimadas, são avaliadas e integradas, resultando em redes de proteínas (FRANCESCHINI et al., 2013). Também possui artigos científicos completos sobre informações de interações que são pesquisados através de mecanismos de mineração de textos e informa sobre estatísticas referente às interações entre as proteínas pesquisadas (FRANCESCHINI et al., 2013).

O STRING visa providenciar um serviço que forneça toda a informação sobre associações funcionais entre proteínas, em um único lugar, atendendo aos desejos dos usuários de ter uma integração e reavaliação facilmente consultadas e navegadas, através de interfaces interativas e intuitivas (SZKLARCZYK et al., 2011). Ele também tem-se especializado em prover dados de forma única e abrangente, ser um entre vários *sites* que armazena interações experimentais, preditas e transferidas entre proteínas junto com interações obtidas através de mineração de texto e na inclusão de uma vasta quantidade de informações adicionais como domínios e estruturas de proteínas (FRANCESCHINI et al., 2013).

2.4 ACESSO À BANCOS DE DADOS BIOLÓGICOS.

Além de armazenar, os bancos de dados em geral precisam fornecer meios de recuperar as informações que armazenam. Existem maneiras diferentes de disponibilizar o acesso aos dados, dependendo do objetivo do banco. Podem ser acessados de forma controlada ou irrestrita, através de acesso local, remoto ou por meio de serviços *web*. Dessa maneira, serão apresentadas as formas de acesso aos bancos do NCBI, através do sistema Entrez, e ao banco de dados STRING por meio de sua *Application Programming Interface* (API).

2.4.1 Entrez

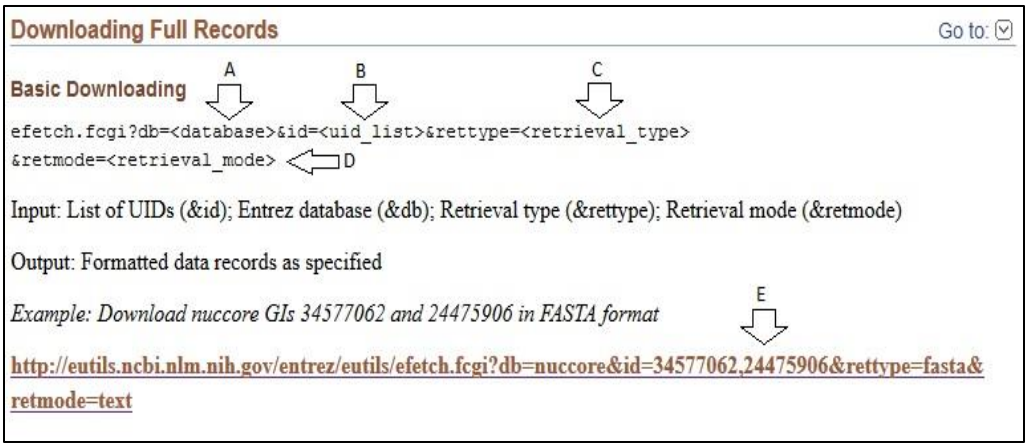
O Entrez é um sistema de recuperação de informações de bancos de dados, onde os acessos a um conjunto de quarenta e dois bancos de dados são disponibilizados através dele, sendo possível pesquisá-los através de texto, por meio de consulta booleana, permitindo o *download* dos dados em formatos diversos e vinculando os registros entre os bancos de dados baseado em relacionamentos confirmados (NCBI, 2014). Esses relacionamentos entre os bancos de dados podem ser entre a sequência e o artigo que a relatou ou entre a sequência da proteína e seu código genético ou com a sua estrutura tridimensional (NCBI, 2014).

O Entrez possui também uma API, denominada *E-utilities*, que possui nove programas que suportam um conjunto uniforme de parâmetros, utilizados para pesquisar, vincular e baixar os dados dos bancos de dados integrados no sistema Entrez (NCBI, 2014). Nessa API são disponibilizadas algumas funcionalidades, através de comandos como o EInfo que retorna estatísticas básicas sobre o banco de dados, o ESearch que retorna os identificadores que foram encontrados baseado nos parâmetros textuais enviados, o EFetch que retorna os dados referentes aos identificadores recebidos, sendo que qualquer comando pode ser enviado para o Entrez processar e retornar através da montagem da *Uniform Resource Locator* (URL) ou pelo *Simple Object Access Protocol* (SOAP) (NCBI, 2014).

Um exemplo de utilização da API *E-utilities* está demonstrado na Figura 1, sendo que este exemplo mostra como utilizar o programa *EFetch* dessa *API* para fazer o *download* de uma sequência completa no formato fasta, através da montagem de uma *URL*. No espaço destacado em A na Figura 1, deve ser colocado o nome do banco de dados do NCBI onde o registro está armazenado. Em B deve ser colocado o identificador, ou uma lista de identificadores separados por vírgulas, referente à consulta desejada. Em C deve ser informado o formato em que o registro será retornado. Em D deve ser colocado o formato em que o arquivo será retornado e, por último, em E, está o exemplo de uma URL completa, montada para retornar registros do

banco de dados *nuccore*, para os identificadores 34577062 e 24475906, onde os resultados estarão no formato fasta e o arquivo de retorno estará no formato texto.

Figura 1 - Exemplo de utilização da API E-utilities com URL.



Fonte: <http://www.ncbi.nlm.moh.gov/>, 2014.

Outro exemplo de utilização da API *E-utilities* está demonstrado na Figura 2, onde é apresentada uma implementação na linguagem Java que faz comunicação com o sistema Entrez pelo protocolo SOAP, e está utilizando o programa ESearch. Para implementar essa comunicação, conforme Figura 2, são necessários os seguintes passos: instanciar o *stub*, destacado em A, criar a requisição, destacado em B, definir o banco de dados, destacado em C, montar os termos de consulta, destacado em D, enviar a requisição e aguardar o resultado, destacado em E e processar os resultados, destacado em F.

Figura 2 - Exemplo de Utilização da API E-utilities com SOAP.

Call ESearch

```
package gov.nih.nlm.ncbi.www.soap.eutils;
import gov.nih.nlm.ncbi.www.soap.eutils.*;
public class Client
{
    public static void main(String[] args) throws Exception
    {
        // search in PubMed Central for stem cells in free fulltext articles
        try
        {
A           EUtilsServiceStub service = new EUtilsServiceStub();
            // call NCBI ESearch utility
            // NOTE: search term should be URL encoded
B           EUtilsServiceStub.ESearchRequest req = new EUtilsServiceStub.ESearchRequest();
C           req.setDb("pmc");
D           req.setTerm("stem+cells+AND+free+fulltext[filter]");
            req.setRetMax("15");
E           EUtilsServiceStub.ESearchResult res = service.run_eSearch(req);
            // results output
            System.out.println("Original query: stem cells AND free fulltext[filter]");
            System.out.println("Found ids: " + res.getCount());
            System.out.print("First " + res.getRetMax() + " ids: ");
F           for (int i = 0; i < res.getIdList().getId().length; i++)
            {
                System.out.print(res.getIdList().getId()[i] + " ");
            }
            System.out.println();
        }
        catch (Exception e) { System.out.println(e.toString()); }
    }
}
```

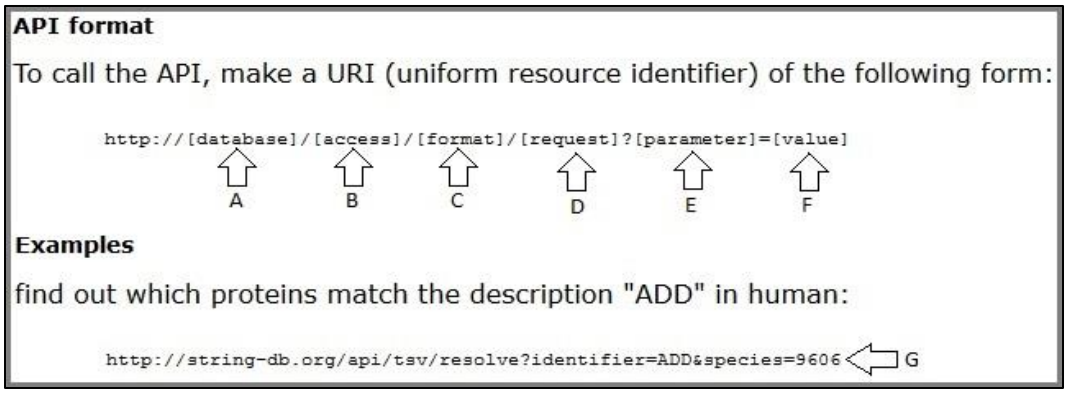
Fonte: <http://www.ncbi.nlm.nih.gov/>, 2014

2.4.2 STRING API

A STRING API é uma ferramenta que visa facilitar a integração do banco de dados STRING dentro ferramentas como o Cytoscape[1] e mecanismos de *workflow* como o Taverna, permitindo o acesso às redes de interação em formatos interpretáveis pelos computadores (JENSEN et al., 2009). A API é chamada através da construção do endereço URL que precisa conter o tipo de requisição, o formato de saída desejado e os dados de entrada, para então o servidor do STRING processar a requisição e retornar os resultados no formato desejado (JENSEN et al., 2009).

Um exemplo de como utilizar a STRING API está demonstrado na Figura 3. Na área destacada em A, na Figura 3, é necessário informar o banco de dados onde o registro está armazenado. Em B, qual o meio de acesso ao banco de dados que está sendo utilizado. Em C, o formato que os dados de resultado serão retornados. Em D, o tipo de requisição que está sendo efetuada. Em E, o parâmetro, ou uma lista de parâmetros separadas pelo caractere &, que está sendo utilizado e, em F, os valores esperados para os parâmetros utilizados. Em G temos a URL montada para efetuar a consulta no banco de dados STRING, através da API, com os resultados retornando no formato *Tab-Separated Values* (TSV), e que a requisição tem que resolver a consulta sobre os parâmetros utilizados. Os parâmetros desse exemplo são as proteínas que possuam o texto *ADD* no seu identificador e que pertençam à espécie *Homo sapiens*.

Figura 3 - Exemplo de utilização da STRING API.



Fonte: <http://www.string-db.org/>, 2014.

2.5 ANÁLISE DE HOMOLOGIA

Ao descobrir um novo gene, ou o gene causador de alguma doença genética, determinar e estudar outras espécies que possuam genes relacionados é um trabalho que interessa aos bió-

[1] *Software* para visualização, criação e manutenção de redes de interações entre proteínas. <http://www.cytoscape.org/>.

logos (LESK, 2008). Os métodos que executam tal comparação devem possuir um equilíbrio entre a sensibilidade e a seletividade para que, respectivamente, selecione sequências mesmo com pouca similaridade em comparação com a original, ao mesmo tempo em que todos os resultados obtidos devem ser verdadeiros (LESK, 2008). Para efetuar essa comparação, a nível de sequência de genes, ferramentas de *Basic Local Alignment Search Tool* (BLAST) são desenvolvidas para que, dada uma sequência de entrada, procurem sequências similares nos registros armazenados nos bancos de dados, avaliando, classificando e fornecendo estatísticas sobre os resultados (AMARAL; REIS; SILVA, 2007).

2.5.1 BLASTp

O BLASTp é um algoritmo que faz a comparação de similaridade entre sequências de aminoácidos das proteínas buscando, a partir de uma sequência origem, regiões de similaridade, sem lacunas, em outras sequências, e conforme outros parâmetros de pesquisa como organismo e banco de dados, reunindo as regiões de similaridade encontradas (LESK, 2008). Um exemplo do funcionamento desse algoritmo está demonstrado na Figura 4, onde o mesmo foi efetuado através do utilitário Entrez, no *site* <http://ncbi.nlm.nih.gov/>, tendo sido consultada a entrada P26367, referente à proteína PAX-6 para a espécie *Homo sapiens*.

No exemplo, conforme apresentado na Figura 4, o resultado mostrado é referente ao organismo *Cynops pyrrhogaster*, destacado em A. Essa consulta mostra a pontuação obtida na comparação, destacado em B, apresenta a probabilidade do resultado ser uma geração aleatória do algoritmo, o *e-value*, destacado em C (*Expect*), entre outras informações estatísticas que possuem a finalidade de pontuar e ordenar os resultados, destacadas em D, E e F. Também apresenta as sequências original e retornada no formato fasta, sendo que nesta parte apresenta três linhas. A primeira, denominada *Query*, destacado em G, é referente à sequência original. Se nesta linha for apresentado o sinal de subtração (-), significa que neste ponto, na sequência original, falta um aminoácido correspondente à mesma posição, indicando que a sequência retornada é maior do que a sequência original (AMARAL; REIS; SILVA, 2007).

A terceira linha, denominada *Sbjct*, destacado em I na Figura 4, apresenta a sequência encontrada e retornada por similaridade. A segunda linha, destacada em H, quando apresenta uma letra ou o caractere *pipe* (|), significa que as sequências original e retornada são exatamente iguais naquela posição, se apresentar o sinal de adição (+) significa que as sequências original e retornada não são iguais porém o aminoácido correspondente é semelhante. Se apresentar um

espaço branco, significa que as sequências não correspondem na posição e a funcionalidade não é similar (AMARAL; REIS; SILVA, 2007).

Figura 4 - Exemplo de resultado do algoritmo BLASTp.

Fonte: <http://blast.ncbi.nlm.nih.gov/Blast.cgi>, 2014.

2.5.2 PSI-BLAST

O algoritmo PSI-BLAST é uma adaptação do algoritmo BLASTp onde é feito um refinamento na etapa inicial da pesquisa, quando está buscando as sequências da base de dados (LESK, 2008). Esse refinamento tem por objetivo identificar padrões conservados, retornando uma quantidade maior de entradas, para após ir reexecutando o algoritmo de BLAST, de forma iterativa, a fim de refinar os resultados e melhorar o padrão reconhecido podendo aumentar tanto a seletividade quanto a precisão da pesquisa (LESK, 2008). Executando esse algoritmo através do utilitário Entrez do NCBI, a apresentação dos resultados na página não é diferente de quando executado o algoritmo BLASTp, demonstrado na Figura 4. No entanto, devido ao processo de identificação de padrões e a iteração do algoritmo, a quantidade de resultados pode ser diferente além do que eles podem ser mais relevantes (LESK, 2008).

Os resultados do algoritmo PSI-BLAST, assim como do BLASTp, podem ser recebidos no formato *Extensible Markup Language* (XML), quando utilizados os recursos da API *E-utilities* do sistema Entrez. Na Figura 5 é demonstrado um arquivo de resultados no formato XML para a execução do algoritmo PSI-BLAST para a proteína ALOX15B para o organismo *Homo sapiens*. Na linha destacada em A na Figura 5, está o número do resultado, baseado na ordenação por *score*, destacado em F. Na linha destacada em B, está o cabeçalho do registro no banco de dados, apresentando o seu identificador no NCBI (gi|85067499), o banco de dados onde o registro foi inserido e o identificador nesse banco de dados (ref|NP_001034219.1). Na linha destacada em C, encontra-se a descrição da proteína. Em D, temos o identificador da proteína no banco de dados onde o registro foi inserido, em E, assim como em F, é apresentado o *score* que o registro obteve em semelhança com a sequência de entrada e, por último, em G, é apresentado o *e-value* que é um valor estatístico que indica a probabilidade do registro ser um falso positivo, ou seja, do registro ser uma geração aleatória do algoritmo.

Figura 5 - Exemplo de resultado do algoritmo PSI-BLAST no formato XML.

Fonte: <http://ncbi.nlm.nih.gov/>, 2014.

2.6 INTEGRAÇÃO DE BANCOS DE DADOS E SISTEMAS

A necessidade de acessar informações e de utilizar ferramentas e sistemas que estão disponíveis em diferentes lugares, sob diferentes plataformas e sistemas operacionais, tem criado um ambiente para o desenvolvimento de ferramentas de *middleware* (BERNSTEIN, 1996). Essas ferramentas permitem que os usuários conectem-se nelas e utilizem das aplicações e bancos de dados por elas integrados tendo por objetivo disponibilizar as informações que as pessoas precisam, quando, onde e como elas precisam (BERNSTEIN, 1996). Uma maneira de integrar os sistemas heterogêneos é oferecendo suporte às interfaces padrões de programação, que tornam a portabilidade das aplicações a uma variedade de tipos de serviços mais fácil e sendo também de fácil aplicação uma vez que os sistemas utilizam bancos de dados, comunicação, apresentação entre outros serviços onde suas telas não fazem parte dessa implementação (BERNSTEIN, 1996).

Os problemas de integração de sistemas heterogêneos também podem ser solucionados com a utilização de protocolos padronizados, que permitem que os sistemas interoperem entre si, ou seja, acessem programas e informações de outros sistemas, enquanto utilizarem o mesmo protocolo padronizado (BERNSTEIN, 1996). Com essa técnica, os sistemas que interoperam podem ser executados em arquiteturas e sistemas operacionais diferentes, desde que essa arquitetura e sistema operacional suportem a interoperabilidade entre sistemas (BERNSTEIN, 1996). Há também as soluções que oferecem, ao mesmo tempo, suporte às interfaces padronizadas de desenvolvimento e também aos protocolos padronizados, sendo então mais completas oferecendo mais opções para o desenvolvimento das integrações dos sistemas (BERNSTEIN, 1996).

2.7 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os conceitos gerais sobre bancos de dados biológicos, como acessá-los, as ferramentas de análise de homologia vinculadas a esses bancos e sobre integração de sistemas heterogêneos. Isto se fez necessário para a compreensão da abrangência da bioinformática e dos mecanismos existentes e que estão disponibilizados aos biólogos moleculares e como esses mecanismos podem ser utilizados no desenvolvimento de experimentos. Na Tabela 1 foi realizada uma comparação entre os bancos de dados referenciados, com o objetivo de apresentar as suas diferenças entre os dados que armazenam, como acessá-los, entre outras informações. No próximo capítulo será abordado o que são *workflows* científicos, apresentando e explicando o funcionamento básico de alguns *workflows* que estão disponíveis para utilização.

Tabela 1 - Comparação entre os bancos de dados biológicos.

	GenBank	**Swiss-Prot**	**STRING**
Tamanho (Nº de Entradas)	171.744.486[2]	545.388[3]	5.214.234[4]
Tipo de Dados	Nucleotídeos, proteínas, estruturas, artigos, taxonomia.	Proteínas, conhecimento, artigos.	Proteínas, estruturas, interações.
Colaboração/Consórcio	INSDC	UniProt	-
Acesso	Entrez, *E-utilities*, SOAP, página *web*	Página *web*, integração EBI	Página *web*, STRING API
Análise Homologia	BLASTp, BLASTx, PSI-BLAST.	BLAST (UniProt)	Consulta por sequência fasta
Integração Interna	Bancos de Dados do NCBI	Não	Stitch
Integração Externa	DDBJ/EMBL-EBI.	EBI/STRING	Swiss-Prot
Interface Integração	Arquivo, API, Interconexões	Interconexões	Interconexões

[2] Fonte: <http://www.ncbi.nlm.nih.gov/genbank/statistics>. Data: 22/05/2014.
[3] Fonte: <http://web.expasy.org/docs/relnotes/relstat.html>. Data: 22/05/2014.
[4] Fonte: <http://string-db.org/>. Data: 22/05/2014.

3 WORKFLOW CIENTÍFICO

Workflows científicos são desenvolvidos para a realização de experimentos *in silico* no campo da bioinformática sendo compostos por outros programas que são chamados, um após o outro, como uma sequência de passos para a realização de um experimento (SILVA; CAVALCANTI; DÁVILA, 2006). A construção de *workflows* torna os experimentos mais fáceis de serem repetidos, tanto para fins de efetuar alguma prova, quanto para fins de aplica-lo em outras situações (SILVA; CAVALCANTI; DÁVILA, 2006).

Os *workflows* também podem capturar as transformações individuais nos dados e analisar os passos, assim como os mecanismos para realiza-los em ambientes distribuídos (GIL et al., 2007). Cada passo do workflow pode ser a chamada de um programa ou a invocação de um serviço *web* sendo ligados de acordo com o fluxo dos dados e com as dependências existentes entre esses dados, além de explorar os complexos processos computacionais para gerenciar seus ciclos de vida e automatizar os seus processos (GIL et al., 2007).

3.1 NET2HOMOLOGY

O programa Net2Homology foi desenvolvido como um *workflow* científico que tem por objetivo realizar a comparação de duas redes de interação proteína-proteína oriundas de dois organismos diferentes (NOTARI, 2012). Essa comparação visa auxiliar na identificação das semelhanças e diferenças entre as funções e estruturas das proteínas que podem ter sido modificadas, ou não, ao longo da evolução das espécies (NOTARI, 2012). Este workflow possui cinco grandes etapas:

i) Seleção de Proteínas: nesta etapa pode-se utilizar um simples arquivo no formato de rede PPI do banco de dados STRING para carregar as proteínas no programa, após selecionando as proteínas desejadas e definindo qual o organismo origem, são os passos A e B conforme a Figura 6;

ii) Consulta de Proteínas por Similaridade: nesta etapa é necessário informar alguns parâmetros como organismo alvo, banco de dados, algoritmo, *e-value* máximo e o formato do arquivo de retorno, com isso cria uma linha de execução para cada proteína selecionada para efetuar o BLAST no NCBI conforme a configuração realizada, são os passos de C à I conforme a Figura 6;

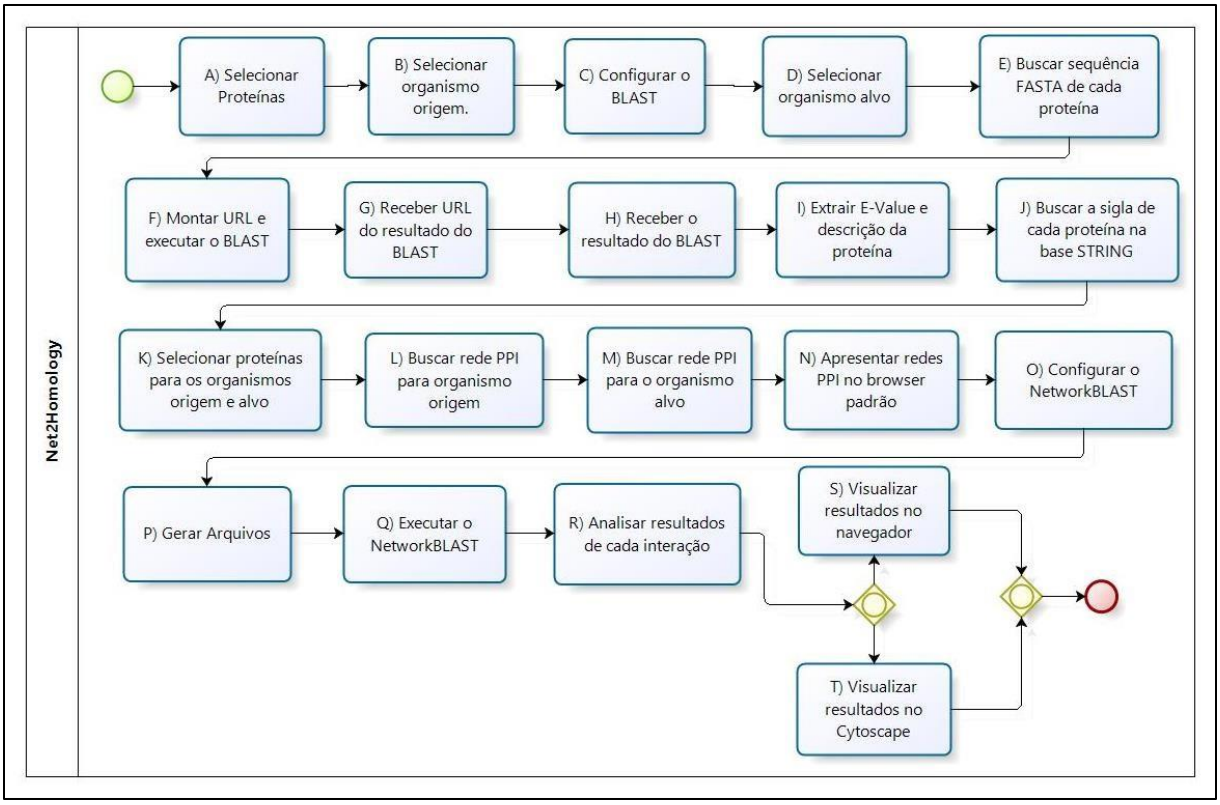
iii) Busca pelas Redes PPI: nesta etapa é feita a busca pelas siglas das proteínas no banco de dados STRING, para todas as proteínas consultadas e retornadas no passo anterior do *workflow*, após devem ser selecionadas as proteínas, uma para o organismo

origem e outra para o organismo alvo para então efetuar a consulta e retornar as redes PPI, são os passos J à N conforme a Figura 6;

iv) Comparação de Redes: nesta etapa é necessário informar alguns parâmetros, onde os principais são a densidade do complexo proteico, taxa de falso negativos em cada rede, *e-value* máximo da sequência similar, o nível de confiabilidade necessário e o número de interações, isso para conseguir utilizar a ferramenta NetworkBlast que resulta em dois arquivos, o primeiro sendo um relatório de execução e o segundo a rede PPI com as proteínas conservadas, sendo que este segundo arquivo pode ser aberto pelo programa Cytoscape, são os passos de O à R e T conforme a Figura 6;

v) Análise de Sistema Biológico: nesta etapa, a rede PPI resultante do processamento da ferramenta NetworkBlast pode ser visualizada, apresentando suas informações e imagens, permitindo ao biólogo efetuar a análise, é o passo S conforme Figura 6.

Figura 6 - Fluxo de execução do workflow Net2Homology



Fonte: autoria própria.

3.2 FERRAMENTAS SEMELHANTES

No campo da bioinformática existem ferramentas desenvolvidas onde cada uma possui um ou mais objetivos. Às vezes esses objetivos podem ser específicos de tal forma que não se encontrem ferramentas com funcionalidades semelhantes. Em outros casos, os objetivos podem

ser abrangentes o que pode tornar a ferramenta mais genérica e por isso podem existir ferramentas similares, abordando os mesmos problemas, ou parte deles, e que atingem objetivos que também podem ser semelhantes.

O *workflow* Net2Homology foi construído para atender um objetivo específico: comparar duas redes de interação proteína-proteína de duas espécies diferentes para verificar as estruturas e funções conservadas durante o processo evolutivo. Portanto, ferramentas que tenham um objetivo parecido não são comuns. Contudo, existem ferramentas dentro da bioinformática desenvolvidas com outros objetivos mas que tratam de problemas semelhantes aos tratados no workflow citado, como integração de informações, consultas aos grandes volumes de dados, entre outros.

3.2.1 BioExtract

O BioExtract é uma plataforma, disponibilizada na *web*, com o propósito de auxiliar os pesquisadores na análise de dados genéticos, oferecendo recursos para o desenvolvimento de *workflows* de bioinformática possuindo uma interface flexível para a execução de consultas aos bancos de dados de proteínas e de nucleotídeos não-redundantes do NCBI e do EBI (LUSHBOUGH; GNIMPIEBA; DOOLEY, 2013). Através dele é possível filtrar os resultados obtidos das consultas nesses bancos de dados e enviá-los como dados de entrada para ferramentas de análise, além de permitir salvar os dados extraídos, integrando-os na ferramenta e utilizando-os como conjunto de dados pesquisáveis (LUSHBOUGH; GNIMPIEBA; DOOLEY, 2013).

O BioExtract além de disponibilizar acesso a ferramentas de análise, está integrado com diversos serviços *web* que efetuam tarefas que auxiliam os pesquisadores, permitindo salvar as tarefas realizadas na ferramenta, em suas etapas, criando *workflows* que podem ser compartilhados, juntamente com os dados extraídos, entre os seus colaboradores (LUSHBOUGH; GNIMPIEBA; DOOLEY, 2013).

3.2.2 SEMEDA

O SEMEDA é uma ferramenta criada com o propósito de gerar um repositório de metadados e semânticas sobre os bancos de dados biológicos, além de criar e armazenar ontologias utilizando vocabulários controlados (KÖHLER; PHILIPPI; LANGE, 2003). Essas ontologias representam significados e, através delas, é possível vincular os metadados para

integrar as informações dos bancos de dados envolvidos (KÖHLER; PHILIPPI; LANGE, 2003).

Em um nível mais técnico, a ferramenta SEMEDA foi desenvolvida em uma arquitetura de três camadas onde o *backend* é um banco de dados relacional que armazena os metadados, semânticas e ontologias (KÖHLER; PHILIPPI; LANGE, 2003). A camada intermediária está implementada na linguagem Java, utilizando as definições e bibliotecas do *Java Server Pages* (JSP) para gerar o HTML dinâmico necessário no *frontend* (KÖHLER; PHILIPPI; LANGE, 2003). Os bancos de dados integrados no SEMEDA podem ser acessados diretamente através do mecanismo de *Java Database Connectivity* (JDBC), ou, nos casos onde os bancos não suportam JDBC, podem ser utilizados módulos externos como BioDataServer, IBMs DiscoveryLinks e DiscoveryHub para utilizar *Structured Query Language* (SQL) homogêneo nas consultas aos bancos de dados (KÖHLER; PHILIPPI; LANGE, 2003).

3.3 CONSIDERAÇÕES FINAIS

Neste capítulo foram apresentados os conceitos gerais sobre *workflows* científicos, além da apresentação das ferramentas Net2Homology, BioExtract e SEMEDA, tendo por objetivo melhorar a compreensão do caso de estudo desse trabalho, o *workflow* Net2Homology. Na Tabela 2, tem uma comparação entre as três ferramentas, comparando as suas finalidades, métodos de integração, entre outras informações. No próximo capítulo serão apresentadas as razões para efetuar esse estudo, problemas da ferramenta e um planejamento de como solucioná-los.

Tabela 2 - Comparação entre as ferramentas.

	Net2Homology	**BioExtract**	**SEMEDA**
Tipo de Ferramenta	*Workflow*	Plataforma	Integrador
Bancos de Dados Integrados	NCBI X STRING	NCBI	Qualquer
Onde Integra	Internamente	NCBI	Internamente
Método de Integração	Cruzamento	Cruzamento	Ontologia e Semântica
Interface de Integração	Arquivos texto	Interconexões do NCBI	Interconexões
Disponível na *Web*	Não	Sim	Sim

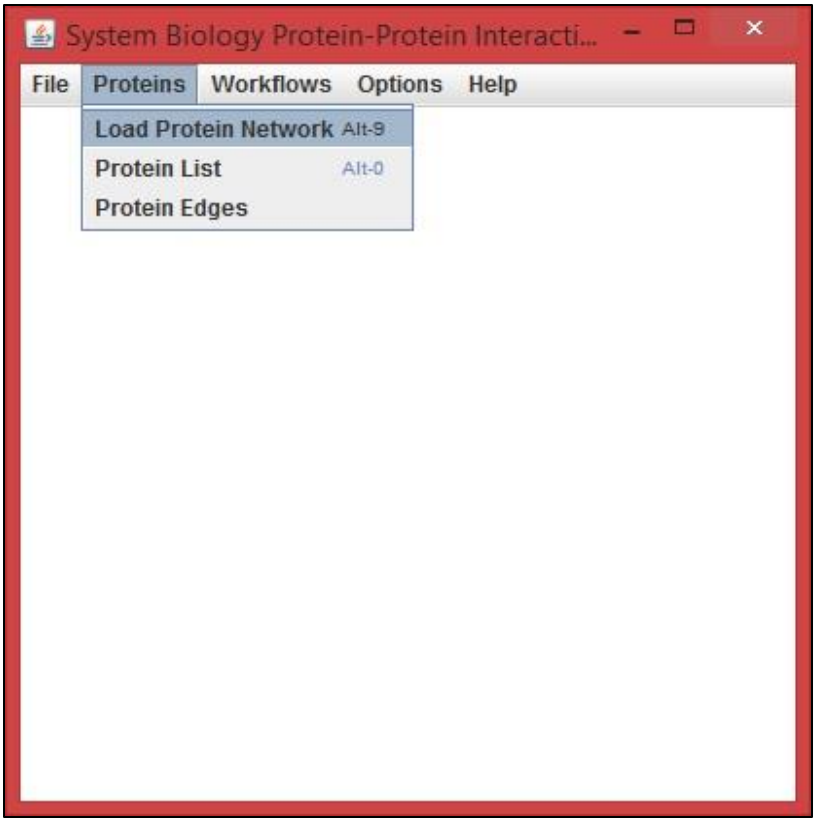
4 PROPOSTA DE SOLUÇÃO

O objeto alvo de estudo neste trabalho é a aplicação Net2Homology, um *workflow* desenvolvido para comparação de redes de interação de proteínas de duas espécies diferentes para auxiliar a identificação das estruturas conservadas durante o processo evolutivo. O fluxo desse *workflow* já foi apresentado na seção 3.1 deste trabalho, portanto agora será apresentada a aplicação em si, suas telas e funcionalidades, seus problemas e um planejamento de como aprimorá-la, no que tange o escopo desse trabalho.

4.1 A APLICAÇÃO NET2HOMOLOGY ATUAL

O programa Net2Homology foi desenvolvido na linguagem Java, versão 1.6 e está disponível para *download* através do *link* <http://www.danlian.com.br/homology/>. Nesse *download* é disponibilizado o arquivo executável da aplicação no formato *Java Archive* (JAR) e para executá-lo é necessária a instalação do *Java Runtime Environment* (JRE) na versão 1.6 ou superior. Ao iniciar a aplicação, é apresentada a tela inicial, demonstrada na Figura 7, que contém a estrutura de menus para acessar a aplicação Net2Homology, sua tela de configurações, entre outras funcionalidades.

Figura 7 - Tela inicial da aplicação Net2Homology.

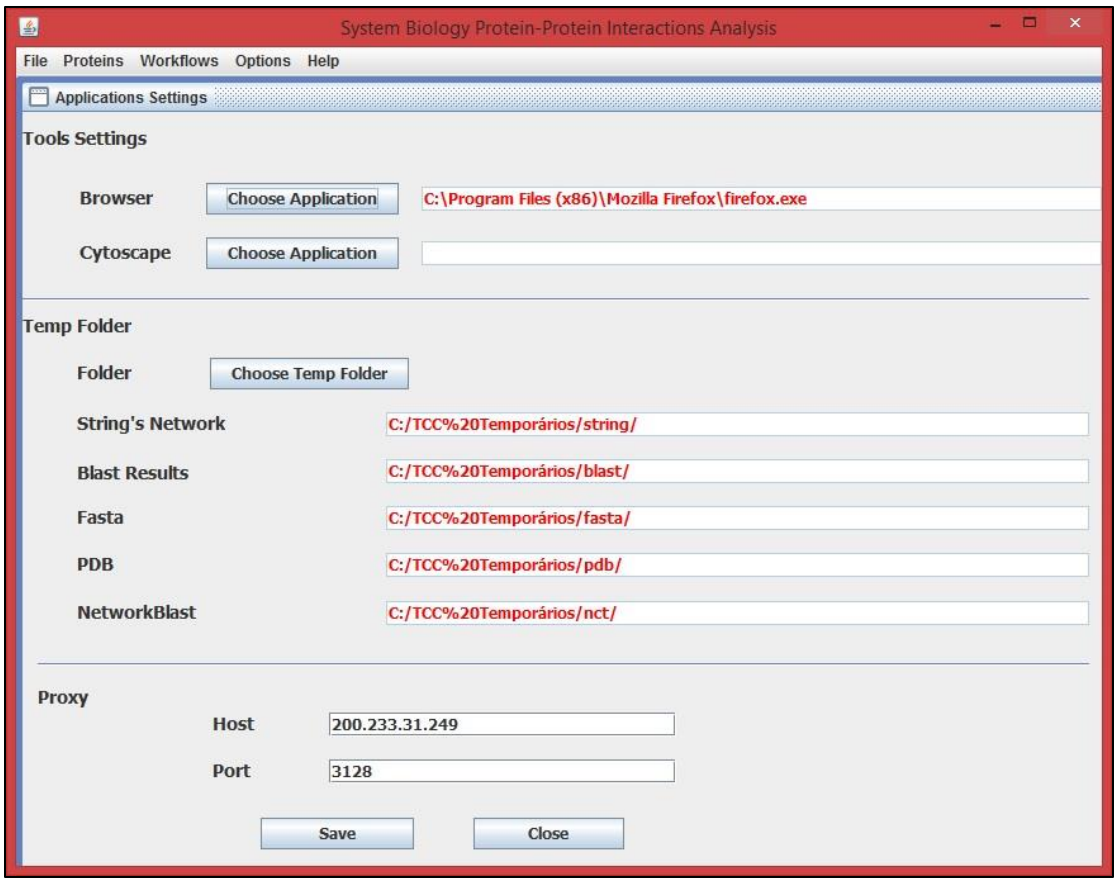


Fonte: autoria própria.

4.1.1 Configuração da Ferramenta

Na primeira vez que a aplicação Net2Homology está sendo utilizada, devem ser efetuadas algumas configurações nela, informando o diretório onde o navegador está instalado, o diretório onde o programa Cytoscape está instalado, uma pasta de trabalho onde os arquivos utilizados durante o processamento serão armazenados e também um endereço de *Internet Protocol* (IP) e uma porta de conexão para um servidor de *proxy*. A tela onde essas informações são cadastradas é acessada através do menu *Options*, opção *Settings* e está demonstrada na Figura 8.

Figura 8 - Tela de configuração da aplicação Net2Homology.



Fonte: autoria própria.

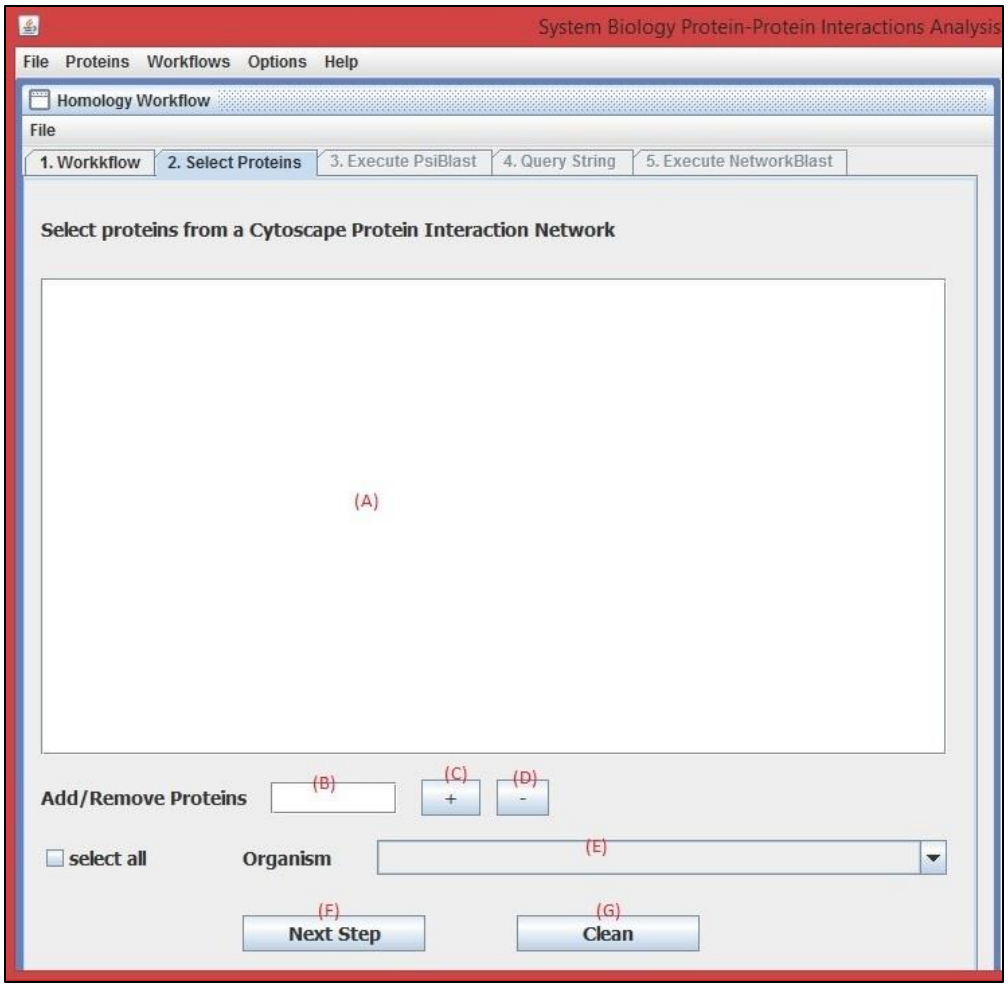
4.1.2 Utilização do *Workflow*

Após iniciar a aplicação e configurar os diretórios e servidor de *proxy*, pode-se começar a utilização do *workflow*, sendo necessário para isso entrar no menu *Workflows* da tela principal e acessar a opção *Homology*. Fazendo isso, é aberta uma tela com a imagem do fluxo do *workflow* Net2Homology e o botão que leva à tela de seleção de proteínas e organismo origem que está demonstrada na Figura 9. Esta tela possui uma lista, destacada em A na Figura 9, que

pode ser atualizada mediante a inserção ou exclusão de informações dela, utilizando os campos destacados em B, C e D.

A lista de proteínas pode já vir preenchida se o programa de carga de proteínas, acessível pelo menu *Proteins*, opção *Load Protein Network* for utilizado para fazer a carga de um arquivo de rede de proteínas no formato TSV que o banco de dados STRING disponibiliza. Nesta tela ainda deve ser selecionado o organismo origem no campo destacado em E na Figura 9, ao qual as proteínas selecionadas pertencem. Após isso, ao clicar no botão *Next Step*, destacado em F, são feitas algumas validações sobre os campos e o *workflow* passa para a próxima tela e também para o próximo passo que é a preparação do BLAST.

Figura 9 - Tela de seleção de proteínas e organismo origem.



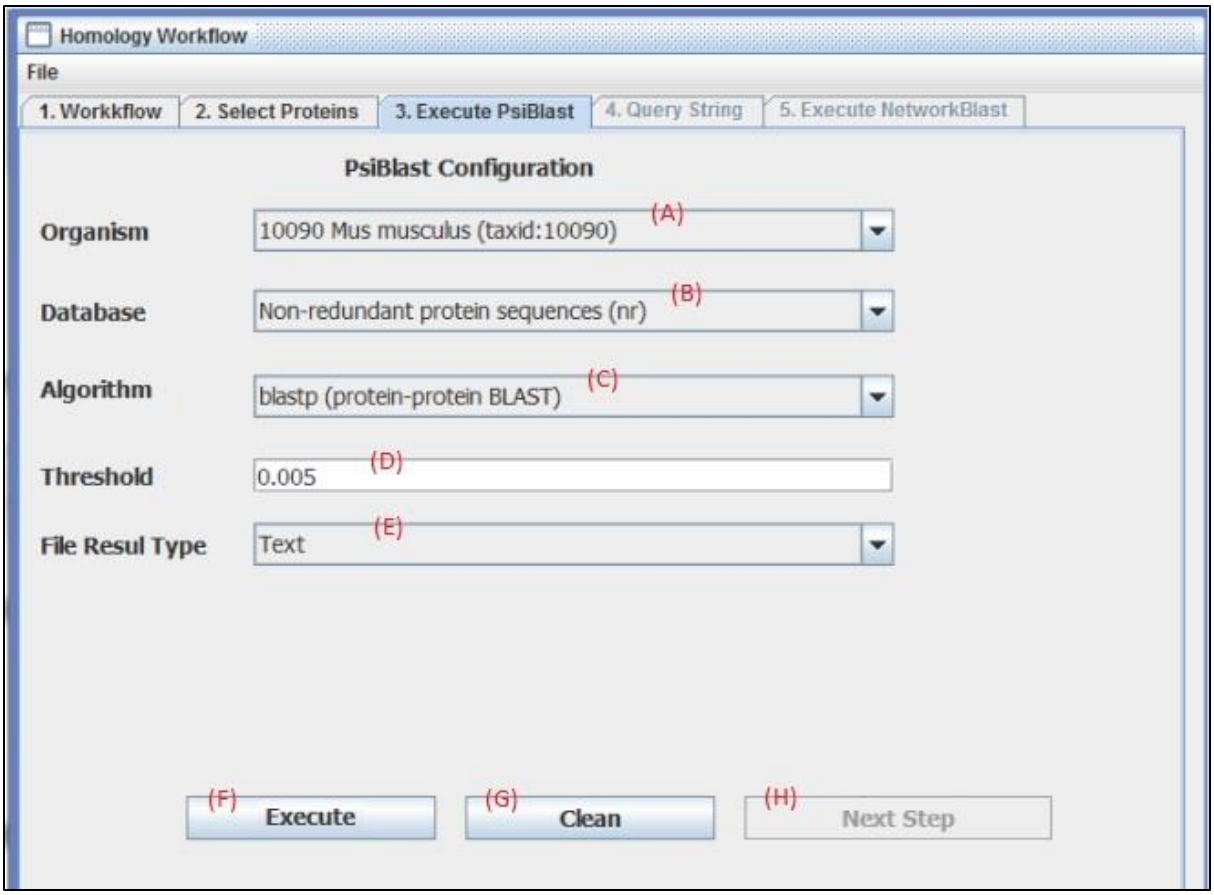
Fonte: autoria própria.

A tela de preparação do BLAST é apresentada na Figura 10. Nesta tela é necessário informar um organismo alvo, no campo destacado em A, qual banco de dados será consultado, no campo destacado em B, o algoritmo de BLAST a ser utilizado, no campo destacado em C, o *e-value* máximo aceito ou, em outros termos, a probabilidade máxima tolerável do resultado do BLAST ser um falso positivo, no campo destacado em D e o tipo de arquivo de resultado,

no campo destacado em E. Ao clicar no botão *Execute*, destacado em F, para cada uma das proteínas selecionadas no passo anterior é criada uma *thread* que irá requisitar a execução do BLAST no sistema Entrez, com base nos parâmetros informados na tela, receber os resultados e vincular a proteína do organismo origem com o melhor resultado obtido para o organismo alvo com base na ordenação que o sistema Entrez faz baseado no *e-value*. Após essa execução, o botão *Next Step*, destacado em H, será habilitado e, se clicado, avançará o *workflow* ao próximo passo e para a tela de seleção e consulta das redes de interação proteína-proteína.

A tela de seleção de proteínas e consulta das redes de interação está demonstrada na

Figura 10 - Tela de configuração do BLAST.



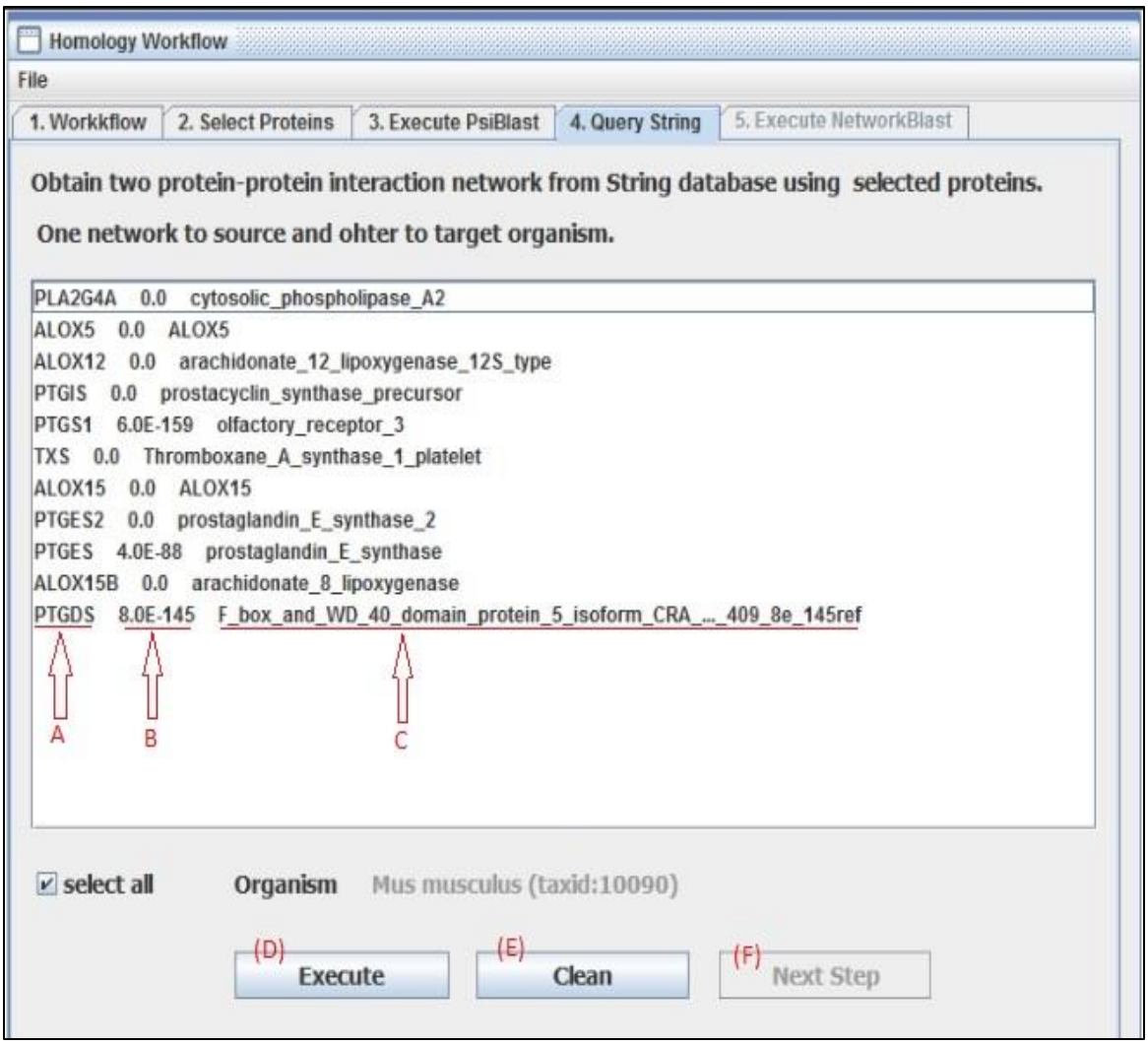
Fonte: autoria própria.

Figura 11. Nela consta uma lista que apresenta três informações, a primeira informação, destacada em A na Figura 11, é a sigla da proteína e a mesma que foi selecionada na primeira etapa do *workflow*. A segunda informação, destacada em B, é o *e-value* do melhor resultado da execução do BLAST para a proteína, efetuado no passo anterior. A terceira informação, destacada em C, é a descrição da proteína equivalente ao mesmo resultado utilizado para apresentar o *e-value* em B. As informações de *e-value* e descrição da proteína são buscadas dos arquivos recebidos com os resultados do BLAST, através de processamento de texto. Por fim,

são consultados os identificadores das proteínas, consultadas e retornadas no BLAST, no banco de dados STRING, através de suas descrições, e, quando encontradas são apresentadas juntamente nessa lista.

Após selecionar as proteínas na lista dessa etapa, é necessário clicar no botão *Execute*,

Figura 11 - Tela de seleção de proteínas e consulta de redes de interação proteína-proteína.



Fonte: autoria própria.

destacado em D para iniciar a consulta das redes de interação proteína-proteína no banco de dados STRING, através dos identificadores encontrados. Ao finalizar a consulta, as imagens e informações sobre as redes podem ser visualizadas no navegador cujo qual foi informado na tela de configurações da ferramenta, e também o botão *Next Step* é habilitado, permitindo prosseguir ao próximo passo do *workflow* e a próxima tela, a tela de configuração e execução do NetworkBlast. A tela de configuração e execução do NetworkBlast não será demonstrada e explicada porque a mesma não está no escopo desse trabalho.

4.2 PROBLEMA ATUAL

O maior problema encontrado na ferramenta está na consulta dos identificadores das proteínas no banco de dados STRING, feita antes de consultar as redes de interação. Devido ao fato da aplicação realizar processamento de texto sobre o resultado para encontrar a descrição da proteína, nem sempre está encontrando a descrição correta. Na Figura 11, na linha sublinhada destacada em C temos um exemplo de uma descrição que não está correta, onde ao final dela está concatenado o valor do *e-value*, destacado em B, também na Figura 11. Devido a isso, nem sempre a aplicação consegue encontrar o identificador da proteína no banco de dados STRING e por consequência não consegue buscar a rede de interação proteína-proteína.

Além do problema na busca do identificador do banco de dados STRING, a etapa de configuração da ferramenta é obrigatória e pode passar despercebida ao usuário porque ela não é requisitada antes de iniciar o seu uso. Também não tem como ser solicitado isso durante o processo de instalação porque a ferramenta não precisa desse processo, bastando efetuar o *download* e executá-la imediatamente após. Além de despercebida, a configuração pode ser algo difícil de realizar considerando que os usuários alvos podem não ter o conhecimento computacional que isso pode exigir, como por exemplo saber o que é um servidor *proxy* e uma porta de conexão entre outras informações que são necessárias nessa etapa.

4.3 A NOVA APLICAÇÃO

Com base nas funcionalidades da aplicação Net2Homology e nos problemas encontrados, o objetivo desse trabalho é fazer uma adaptação dessa ferramenta, mantendo as suas funcionalidades para que continue atingindo ao objetivo a que se propõe e resolver os problemas que foram encontrados ao utilizá-la.

4.3.1 Seleção de Organismo Origem e Proteínas

A tela de seleção de organismo origem e proteínas é a etapa inicial do *workflow* Net2Homology e deverá ter as seguintes funcionalidades:

a) Campo para selecionar o organismo origem;
b) Uma lista de seleção de proteínas;
c) Campo para inserir e/ou remover proteínas da lista de seleção;
d) Funcionalidade de carregar e selecionar proteínas originadas de arquivos com dados no formato TSV e TSV sem cabeçalho, que a aplicação atual já faz;
e) Botão para ir ao próximo passo do *workflow*;

f) Botão para limpar os dados selecionados;

g) Verificar se o organismo origem foi informado;

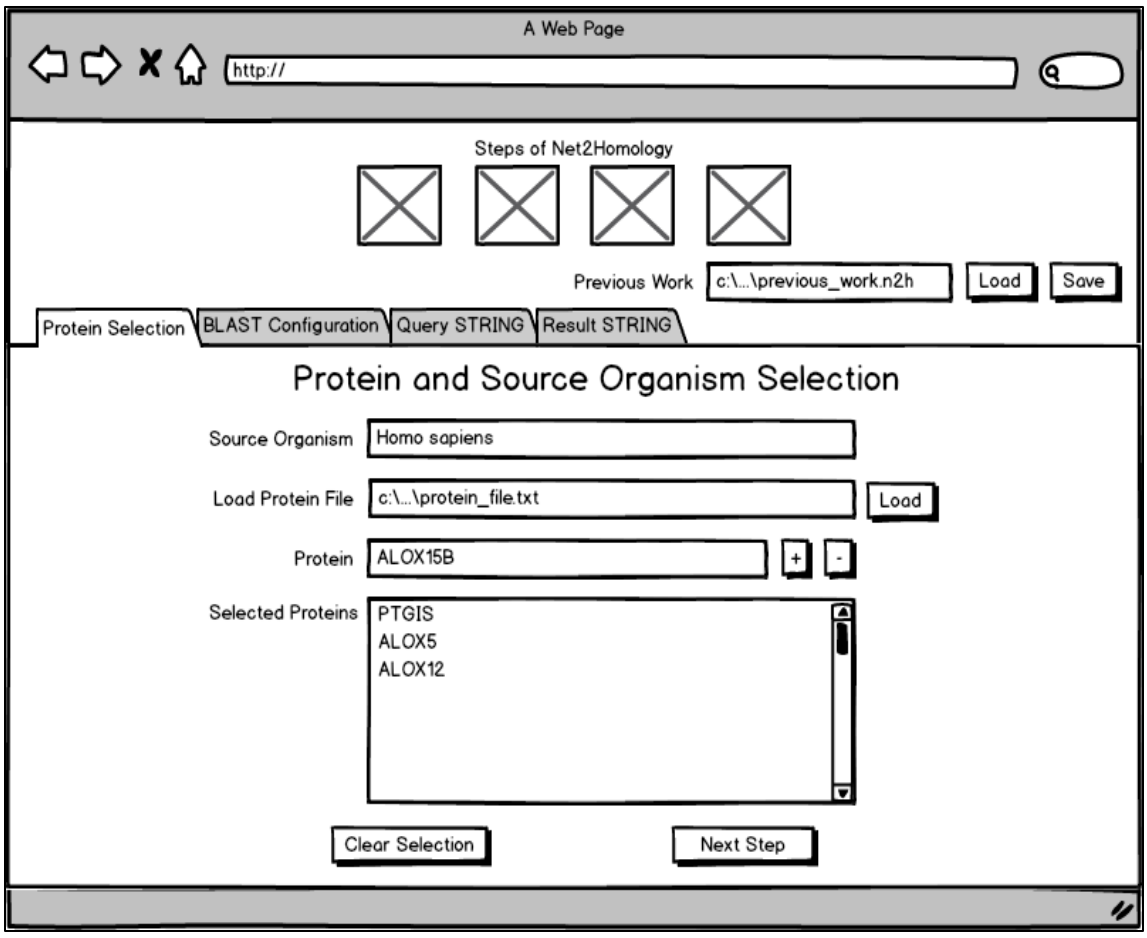
h) Verificar se foi selecionada ao menos uma proteína;

i) Apresentar mensagens informativas sobre problemas ocorridos em validações e processos efetuados.

j) Só permitir prosseguir ao próximo passo se todas informações foram validadas.

Para atender às funcionalidades citadas, a tela de seleção de proteínas e organismo origem deverá ser semelhante ao esboço ilustrado na Figura 12.

Figura 12 - Esboço da tela de seleção de proteínas e organismo origem.



Fonte: autoria própria.

4.3.2 Configuração do BLAST

A tela de configuração do BLAST é a segunda etapa do *workflow* Net2Homology e deve ter os seguintes campos e funcionalidades:

a) Campo para selecionar o organismo alvo;

b) Campo para selecionar o banco de dados de proteínas de uma lista de opções fixa;

c) Campo para selecionar o algoritmo de BLAST de uma lista de opções fixa;

d) Campo para informar o *e-value* máximo tolerável;

e) Botão para executar o BLAST no NCBI pelo sistema Entrez, conforme parâmetros da tela;

f) Botão para limpar as informações da tela;

g) Botão para avançar à próxima etapa do *workflow*;

h) Verificar se o organismo alvo foi informado;

i) Receber resultado do BLAST em arquivo no formato XML;

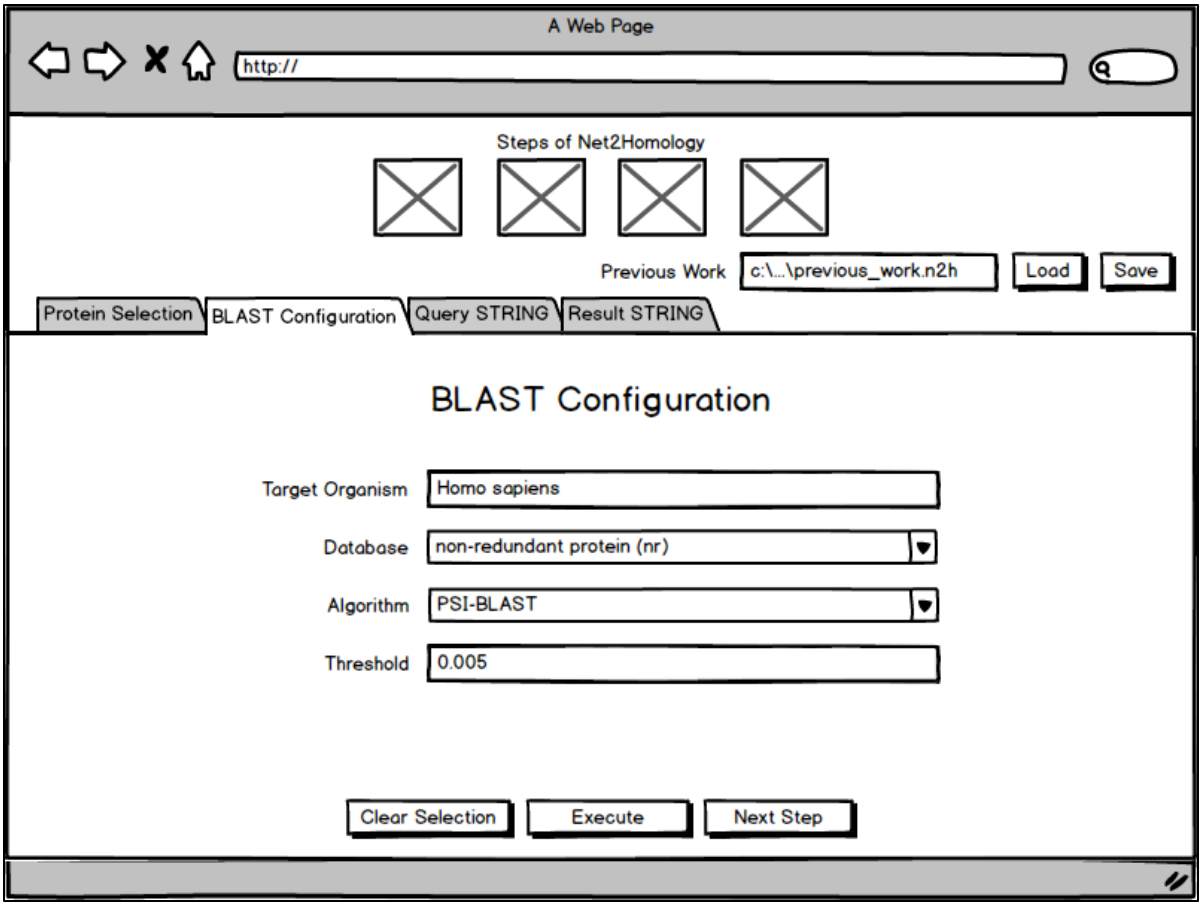
j) Criar uma linha de execução para cada proteína selecionada no primeiro passo do *workflow*;

k) Só prosseguir ao próximo passo após executar o BLAST;

l) Apresentar mensagens informativas referente aos problemas em validações e processos efetuados.

Para atender às funcionalidades citadas, a tela de configuração do BLAST deverá ser semelhante ao esboço ilustrado na Figura 13.

Figura 13 - Esboço da tela de configuração do BLAST.



Fonte: autoria própria.

4.3.3 Consulta ao STRING

A tela de consulta ao banco de dados STRING é a terceira etapa do *workflow* Net2Homology e deverá ter as seguintes funcionalidades:

a) Apresentar as proteínas selecionadas no primeiro passo do *workflow*;

b) Vincular a proteína de melhor *score* do resultado da execução do BLAST;

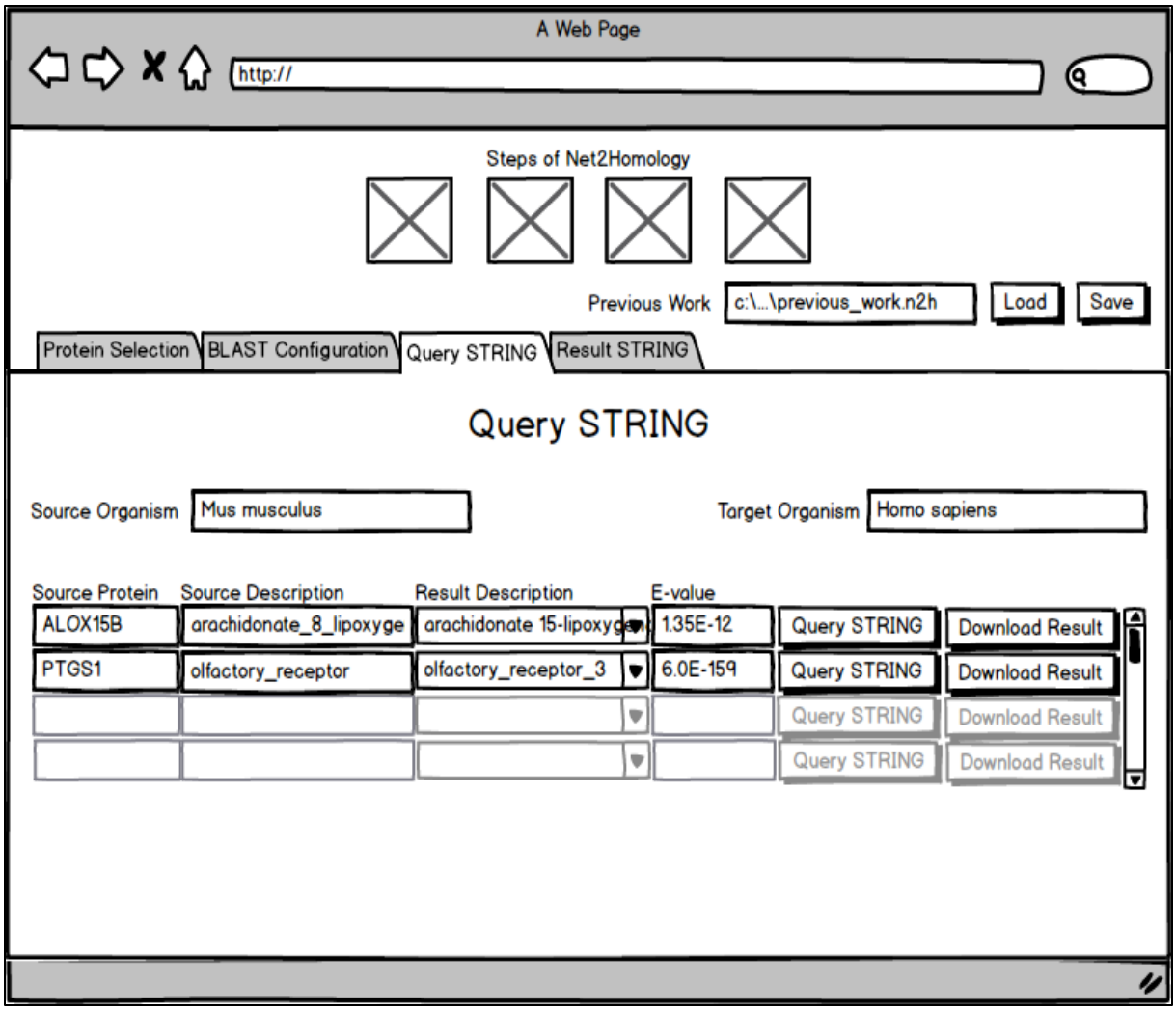
c) Permitir alterar a proteína vinculada por qualquer outra que tenha retornado do BLAST para a proteína origem;

d) Permitir efetuar o *download* dos resultados do BLAST;

e) Efetuar a consulta das redes de interação proteína-proteína no banco de dados STRING;

Para atender às funcionalidades citadas, a tela de consulta ao banco de dados STRING deverá ser semelhante ao esboço ilustrado na Figura 14.

Figura 14 - Esboço da tela de consulta ao banco de dados STRING.



Fonte: autoria própria.

4.3.4 Visualização dos Resultados da consulta ao STRING

A tela de visualização dos resultados da consulta ao banco de dados STRING é nova em relação à aplicação atual. No entanto, o passo referente à essa tela não acontecia dentro da ferramenta, mas sim no navegador informado nas configurações. Portanto essa tela deverá ter as seguintes funcionalidades:

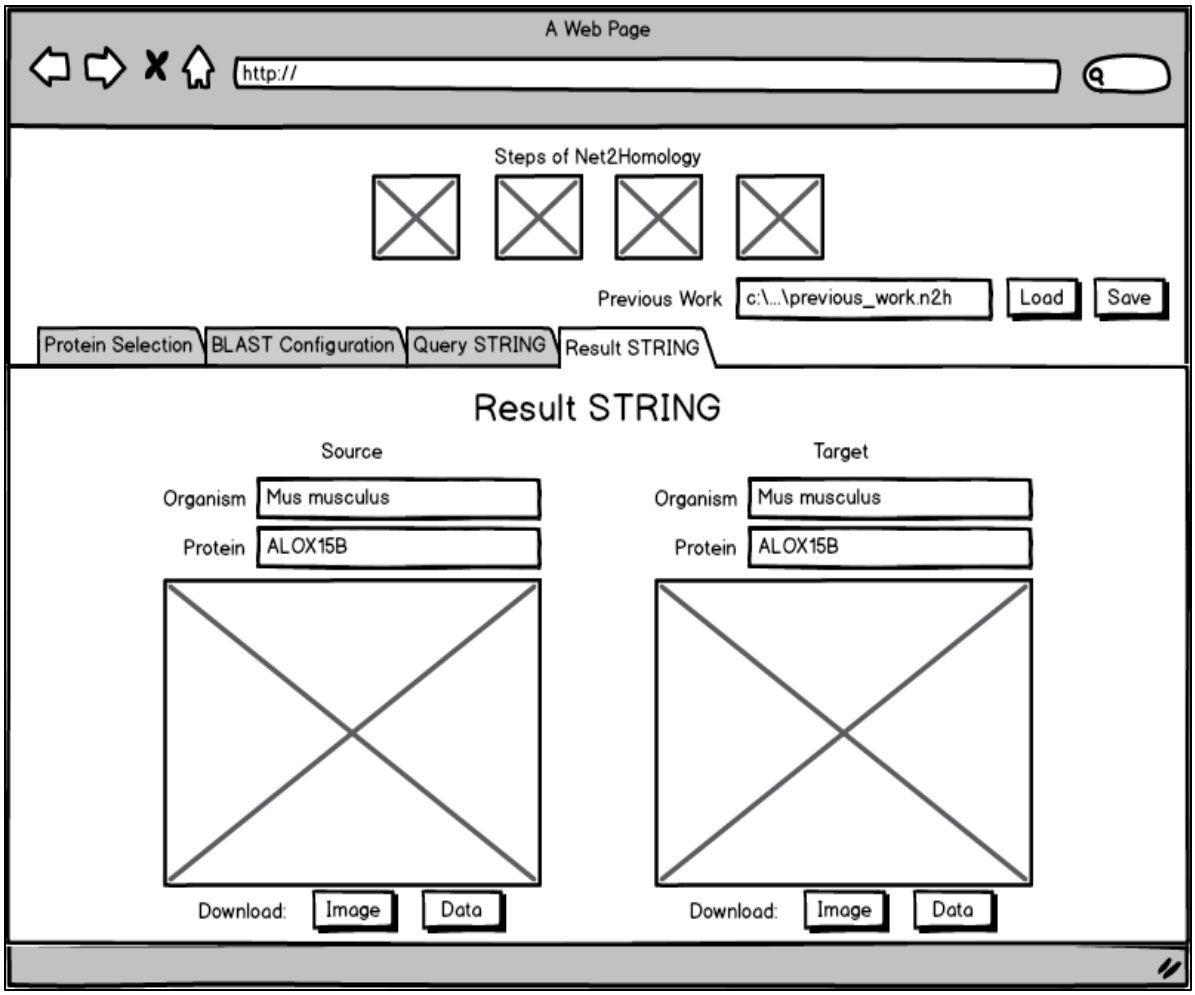
a) Apresentar as imagens das redes de interação proteína-proteína da proteína do organismo origem e da proteína do organismo alvo;

b) Deverá permitir o *download* das imagens das redes de interação proteína-proteína;

c) Deverá permitir o *download* dos dados das redes de interação proteína-proteína.

Para atender essas funcionalidades, a tela de visualização dos resultados da consulta ao STRING deverá ser semelhante ao esboço ilustrado na Figura 15.

Figura 15 - Esboço da tela de visualização do resultado da consulta ao STRING.



Fonte: autoria própria.

4.3.5 Integração NCBI x STRING

O procedimento que busca os identificadores das proteínas selecionadas no banco de dados STRING precisa ser aprimorado, a fim de diminuir a quantidade de redes de interação que não podem ser consultadas por problemas nessa etapa. Para esse aprimoramento, serão necessárias as seguintes funcionalidades:

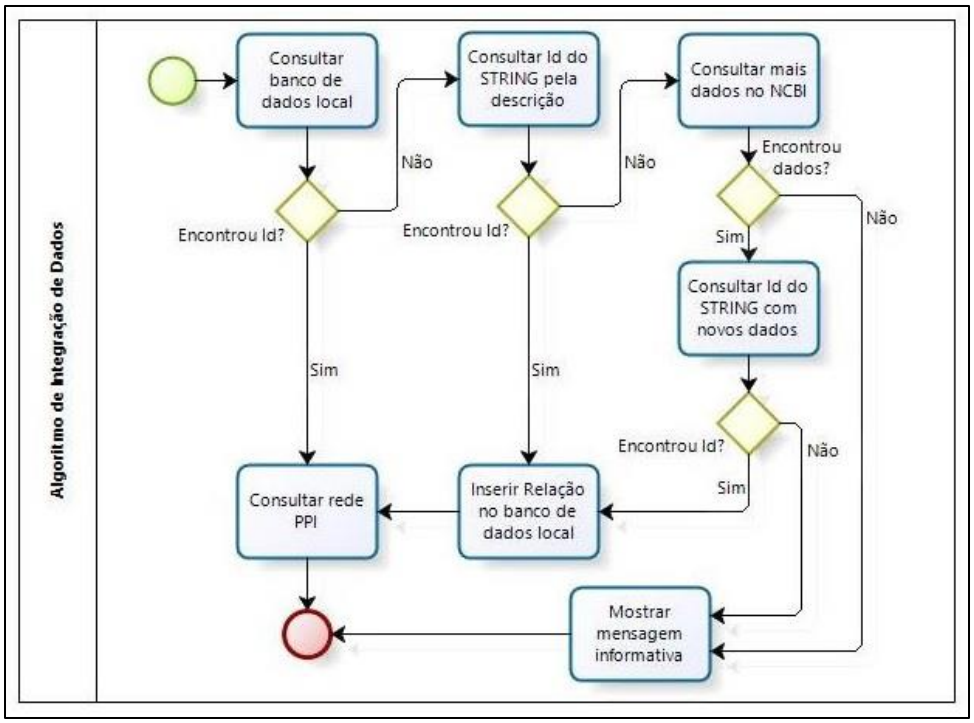
a) Armazenar a relação (identificador da espécie, identificador do NCBI, identificador do STRING) em banco de dados local;

b) Consultar o identificador do STRING, pela STRING API, utilizando a descrição da proteína;

c) Consultar o NCBI novamente, por BLAST, ESearch e EFetch, para buscar mais informações que identifiquem a proteína para consultar o STRING novamente.

Esse procedimento de consulta acontece entre os passos de consulta ao STRING e visualização dos resultados do STRING. O algoritmo que fará esse procedimento terá o fluxo ilustrado na Figura 16.

Figura 16 - Fluxo do algoritmo de busca do identificador no banco de dados STRING.

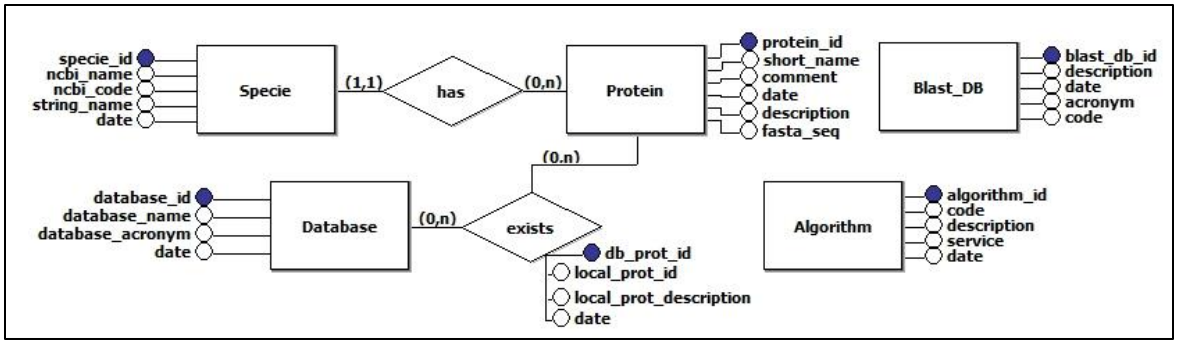


Fonte: autoria própria.

Como demonstrado na lista de funcionalidades desse processo e no fluxo do algoritmo que o executará, a integração de dados entre os bancos do NCBI e do STRING será feita através de banco de dados local, onde serão armazenadas as referências diretas entre os identificadores das proteínas referente às duas fontes de dados. Também foi escolhido esse método pelo fato

de ser disponibilizado pelo banco de dados STRING um arquivo de referências diretas entre os identificadores das proteínas no STRING e os identificadores das mesmas proteínas em outros bancos de dados. Os dados desse arquivo estão no formato TSV e seu *download* pode ser feito através do *link* <http://string-db.org/newstring_download/protein.aliases.v9.1.txt.gz>. Da mesma forma que para as proteínas, para os organismos também existe disponibilizado um arquivo em formato TSV com as referências diretas entre o banco de dados STRING e os bancos de dados do NCBI através do *link* < http://string-db.org/newstring_download/species.v9.1.txt>. Esses arquivos serão importados em estruturas de um banco de dados local a ser criado. O diagrama entidade-relacionamento levantado para esse requisito está ilustrado na Figura 17.

Figura 17 - Diagrama entidade-relacionamento da aplicação Net2Homology.



Fonte: autoria própria.

4.3.6 Implementação

A nova aplicação Net2Homology será uma ferramenta disponível na *web*, acessando-a pelo navegador, com o objetivo de não ser mais necessário ao usuário configurá-la antes do seu uso. Ela terá a estrutura definida pela arquitetura lógica *Model-View-Controller* (MVC) devido à utilização do *framework* JSF que é baseado nessa arquitetura. Um detalhamento maior sobre as arquiteturas física e lógica é feito nas próximas seções.

4.3.6.1 Arquitetura Física

As ferramentas disponíveis que serão utilizadas nesse trabalho, para desenvolvê-lo na camada *web*, atendendo aos requisitos descritos são:

a) Servidor de banco de dados PostgreSQL9.3: o banco de dados citado na seção 4.3.5, para aprimoramento da integração NCBI com STRING;

b) Servidor *web* Apache Tomcat 7.0: o servidor *web* J2EE que irá disponibilizar a aplicação na *internet*, requisitado pelo proprietário da aplicação.

c) *Framework* JSF 2.2: camada que gera o HTML final para apresentação no navegador;

d) Biblioteca de componentes PrimeFaces 5.0: biblioteca para auxílio na criação das interfaces da aplicação;

e) *Java Development Kit* (JDK) 1.8: a aplicação será desenvolvida na linguagem Java, versão 1.8;

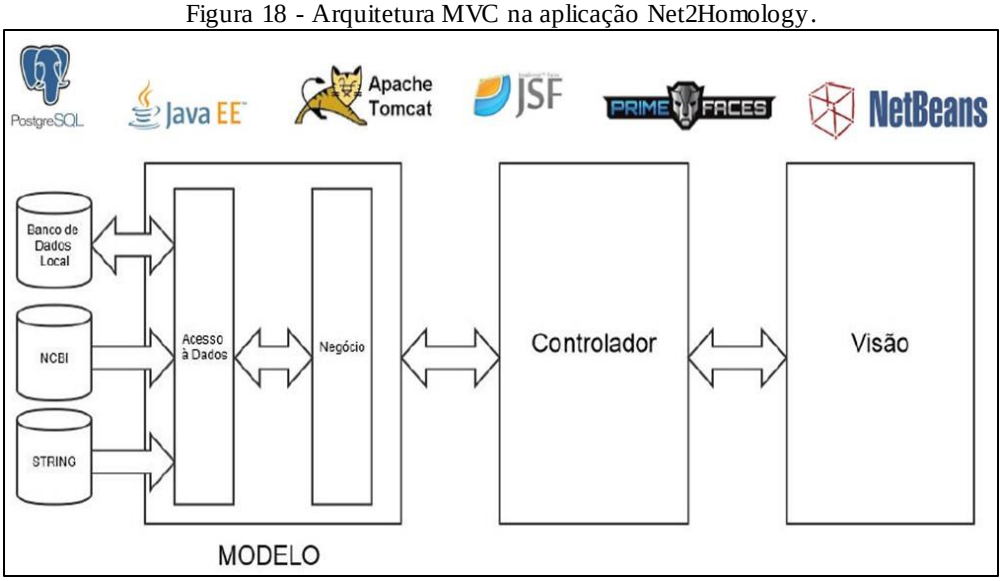
f) IDE de desenvolvimento NetBeans 8.0: o programa para desenvolvimento da aplicação.

4.3.6.2 Arquitetura Lógica

A arquitetura lógica da aplicação seguirá o modelo MVC, pela utilização do *framework* JSF 2.2 que está implementado nessa arquitetura. A camada controladora será o JSF, gerando o HTML para a visualização e enviando as requisições ao servidor. A camada de visão será construída em HTML interpretável pelo JSF que irá gerar o HTML final para visualização. Para auxiliar a construção do HTML da camada de visão, será utilizada a biblioteca de componentes PrimeFaces.

A camada de modelo, prevista no MVC como a camada de dados, será dividida em outras duas camadas, uma camada de negócio, onde as regras de negócio da aplicação serão desenvolvidas e com a qual a camada de controle fará comunicação. A segunda camada que dividirá a camada de modelo do MVC será a camada de acesso a dados, que irá conter os meios de acesso ao banco de dados local e aos serviços *web* do sistema Entrez e do banco de dados STRING. Uma ilustração dessa arquitetura está demonstrada na Figura 18.

Figura 18 - Arquitetura MVC na aplicação Net2Homology.



Fonte: autoria própria.

4.3.7 Testes

Após o desenvolvimento da aplicação, serão realizados testes das funcionalidades que envolvam a comunicação com o sistema Entrez e também com o banco de dados STRING, comparando os resultados obtidos durante a execução da aplicação com os resultados que são obtidos ao realizar as consultas através de seus sistemas *web*, utilizando dos mesmos parâmetros de filtro. Esse teste será considerado satisfatório quando os resultados obtidos em ambos os processos forem iguais.

Para testar a nova integração entre os bancos de dados do NCBI e o banco de dados STRING, serão realizadas consultas cruzadas. Nessas consultas cruzadas, as proteínas obtidas ao consultar o banco de dados STRING serão informadas na ferramenta de BLAST do NCBI, disponível na página *web* do sistema Entrez, utilizando de suas sequências fasta e dos respectivos organismos como parâmetros de filtros. A integração terá um resultado satisfatório caso, no relatório de resultados do BLAST executado nesse momento, forem encontradas as proteínas que, durante a execução da aplicação, foram retornadas das consultas ao NCBI ou selecionadas na etapa inicial do *workflow*.

Também serão realizados testes de homologação da aplicação na *web* sendo testadas as suas telas e funcionalidades visuais e de negócio ao utilizar a aplicação no navegador. Para esses testes, a aplicação será executada nos navegadores Mozilla Firefox versão 30 e Internet Explorer versão 11, sendo satisfatório se os componentes e funcionalidades não apresentarem problemas durante a execução desses testes.

4.4 CONSIDERAÇÕES FINAIS

Neste capítulo foi abordado sobre a aplicação Net2Homology, suas funcionalidades, seus problemas e sobre uma possível solução para eles, a ser implementada e testada, conforme os requisitos levantados e detalhados. No próximo capítulo será detalhada como a implementação das soluções propostas foram desenvolvidas, testadas e implantadas.

5 DESENVOLVIMENTO

Conforme o planejamento efetuado nas etapas anteriores deste trabalho, verificou-se que a reconstrução da ferramenta Net2Homology, partindo do zero, seria mais eficiente considerando o fato que não seria feita apenas a troca da plataforma da ferramenta, de *desktop* para *web*, mas também a adaptação do algoritmo de integração entre os bancos de dados do NCBI e o banco de dados STRING. Nas próximas seções serão detalhadas algumas etapas do processo de desenvolvimento.

5.1 PREPARAÇÃO DO AMBIENTE

A primeira etapa realizada no desenvolvimento foi a preparação do ambiente, quando foi montado o *framework* de desenvolvimento escolhido na etapa de planejamento. Nessa preparação, precisou ser feito o *download* e a instalação da *Java Virtual Machine* (JVM) nas versões 7 e 8, do servidor de banco de dados PostgreSQL 9.3, do servidor *web* Apache Tomcat 7, do *framework* JSF 2.2, da biblioteca de componentes Primefaces 5.0 e da IDE de desenvolvimento NetBeans 8.0.

Nesta etapa também foi necessário o download das APIs eutils_axis2 (*e-utilities*) para a realização de consultas ao Entrez utilizando de seus *webservices*, Jdom 2.0.5 para processamento de dados no formato XML, axis2 (Apache) para o correto funcionamento da API eutils_axis2, postgresql9.3 para configuração e utilização do driver JDBC do banco de dados PostgreSQL entre várias outras que são anexadas automaticamente ao projeto quando as APIs citadas anteriormente são anexadas.

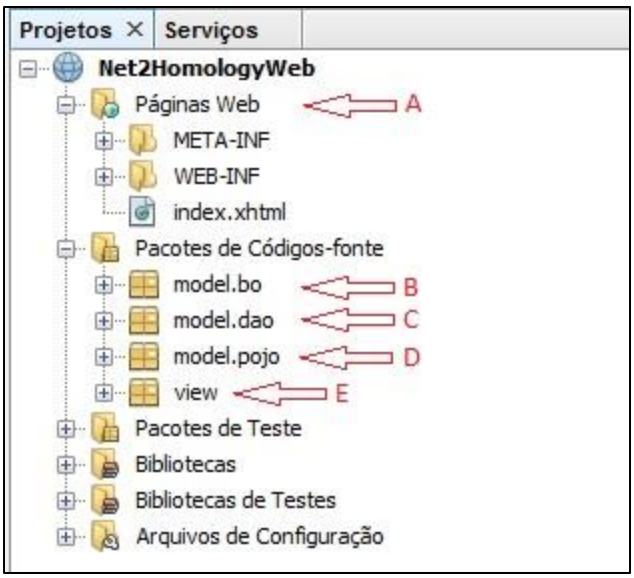
5.2 O PROJETO

Após a execução dos *downloads* e da instalação das ferramentas necessárias ao desenvolvimento, utilizando a IDE NetBeans, foi criado um novo projeto *web*, denominado Net2HomologyWeb. Neste projeto foram criadas as *packages* e pastas para agrupar as classes e arquivos conforme suas funcionalidades. A Figura 19 ilustra a estrutura das *packages* e arquivos do projeto Net2HomologyWeb.

Conforme a Figura 19, em A está identificada a pasta onde são armazenados os arquivos xHTML das páginas da aplicação que está sendo desenvolvida. No caso do projeto Net2HomologyWeb foi suficiente apenas um arquivo xHTML, o index.xhtml, onde todas as telas da aplicação foram implementadas. Em B está indicada a *package* bo (*Business Object*) que está incluída dentro da *package* model. Nessa *package* foram adicionadas todas as classes que fazem

algum tipo de processamento na ferramenta Net2Homology, possuindo então as suas regras de negócio. Em C está indicada a *package* dao (*Data Access Object*), também incluída na *package* model. Nessa *package* foram incluídas todas as classes que fazem acesso aos dados que a aplicação manipula, sejam dados em arquivos, dados em bancos de dados locais ou mesmo dados em bancos de dados remotos. Em D está indicada a *package* pojo (*Plain Old Java Object*), também contida na *package* model. Nessa package foram adicionadas todas as classes dos objetos que são utilizados em todas as camadas da aplicação, funcionando como um simples conjunto de dados. Em E está indicada a *package* view onde foram adicionadas a classe *bean* e os conversores das classes pojos.

Figura 19 - Estrutura de *packages* e pastas do projeto Net2HomologyWeb.



Fonte: autoria própria.

5.3 CODIFICAÇÃO

A codificação da aplicação se fez, inicialmente, pela construção do arquivo index.xhtml tendo por objetivo criar as interfaces da aplicação logo no início do desenvolvimento, para ser possível realizar os testes com telas desde o começo do projeto. Após isso, foi criado o banco de dados da aplicação utilizando o programa pgAdmin[5] que é a ferramenta de administração do banco de dados PostgreSQL. Em seguida foram executados os *scripts* para criação das tabelas, chaves primárias, chaves estrangeiras e sequencias da aplicação. Com as estruturas criadas no banco de dados, foram executados alguns *scripts* para a inserção de dados em algumas tabelas, como a tabela de espécies, que precisam ter informações para o funcionamento da aplicação.

[5] Programa para administração do banco de dados PostgreSQL. <http://www.pgadmin.org>.

Após a criação do banco de dados foi realizado o desenvolvimento das funcionalidades da aplicação, tela por tela, seguindo a mesma ordem das etapas do *workflow*. O desenvolvimento da aplicação dessa forma foi necessário porque, justamente por se tratar de um *workflow*, a realização de testes em uma de suas etapas é diretamente afetada pelas etapas anteriores à ela, quando há, existindo uma dependência funcional.

Em um nível mais baixo, a codificação da aplicação seguiu quase sempre a mesma ordem que foi inicialmente a criação ou alteração da classe *bean*, desenvolvendo os tratamentos de interface, depois, quando necessário, a criação ou alteração das classes de negócio, desenvolvendo os processos conforme as regras da aplicação. Após isso, foi feita a criação ou alteração das classes de acesso aos dados, desenvolvendo os processos para conexão, consulta e retorno de dados aos arquivos e bancos de dados utilizados. Em meio a todas essas alterações, as classes pojos foram sendo criadas ou alteradas, para atender aos requisitos de todas as camadas da aplicação.

Na conclusão da codificação, se realizaria a importação no banco de dados local do arquivo de referências das proteínas do banco de dados STRING com outros bancos de dados, conforme relatado na seção 4.3.5 desse trabalho, contudo essa tarefa não foi possível por conta dos seguintes motivos:

a) Os dados do arquivo não possuíam informações relevantes para a realização de consultas nos bancos de dados do NCBI, ocasionando que a maioria das consultas não retornavam nenhuma informação. Após setenta e duas horas de execução do *script* de importação, tendo já processado milhares de linhas, pouco mais de duzentas e oitenta proteínas haviam sido incluídas no banco de dados local;
b) O arquivo possui pouco mais de três gigabytes de tamanho e, devido a esse volume associado a uma requisição HTTP para consultar os bancos de dados do NCBI para cada linha do arquivo, tornou-se inviável aguardar a conclusão do *script* de importação considerando a data de início de execução além do tempo disponível a partir dessa data para a conclusão do mesmo.

5.4 TESTES E HOSPEDAGEM

Durante todo o processo de codificação, testes das telas e funcionalidades da aplicação foram executados, para identificar e corrigir, quando necessário, problemas nos campos ou nos processos desenvolvidos. Para fins de validação da ferramenta, testes das funcionalidades foram realizados após as conclusões de cada uma das etapas, assim como ao final do desenvolvimento da aplicação, verificando se os processos estavam de acordo com as regras de negócio. Durante

esses testes, nenhum grande problema foi identificado que tivesse causado algum transtorno que impactasse no desenvolvimento ou na validação da ferramenta.

A hospedagem da aplicação no servidor não ocorreu porque esse servidor apresentou problemas onde, mesmo seguindo as instruções fornecidas pela documentação, a execução delas não surtia o efeito esperado para concluir a hospedagem. A fim de resolver essa situação, foi aberta uma solicitação no sistema do servidor de hospedagem, informando do ocorrido e, até a data de conclusão desse trabalho esse problema não havia sido solucionado, o que ocasionou a sua não hospedagem na *web*.

5.5 CONSIDERAÇÕES FINAIS

Neste capítulo foi descrito como o desenvolvimento do *workflow* Net2Homology na plataforma *web* foi realizado, apresentando as etapas de preparação do ambiente, criação do projeto, codificação, testes e hospedagem. Na Tabela 3 é apresentada uma comparação entre as funcionalidades das duas versões da ferramenta Net2HomologyWeb, uma na plataforma *desktop* e outra na plata-forma *web*. No próximo capítulo será apresentado um estudo de caso, mostrando as telas da aplicação durante sua utilização com dados reais e uma comparação dos resultados obtidos nessa aplicação na plataforma *web* com os resultados que eram obtidos na versão anterior, na plataforma *desktop*.

Tabela 3 - Comparação de funcionalidades entre as versões da ferramenta Net2Homology.

	Desktop	***Web***
Seleção de Proteínas	Sim	Sim
Seleção de Organismo	Sim	Sim
***Upload* de Arquivo de Proteínas**	Sim	Sim
***BLAST* em Linhas de Execução**	Sim	Sim
Download* de Relatórios de *BLAST	Não	Sim
Seleção de Resultado de BLAST para Consulta da Rede PPI	Não	Sim
Banco de Dados Local para Integração	Não	Sim
Consulta das Redes PPI	Sim	Sim
Apresentação das Imagens na Ferramenta	Não	Sim
***Download* das Redes PPI**	Não	Sim
Todas Etapas do *Workflow* Desenvolvidas	Sim	Não

Fonte: autoria própria.

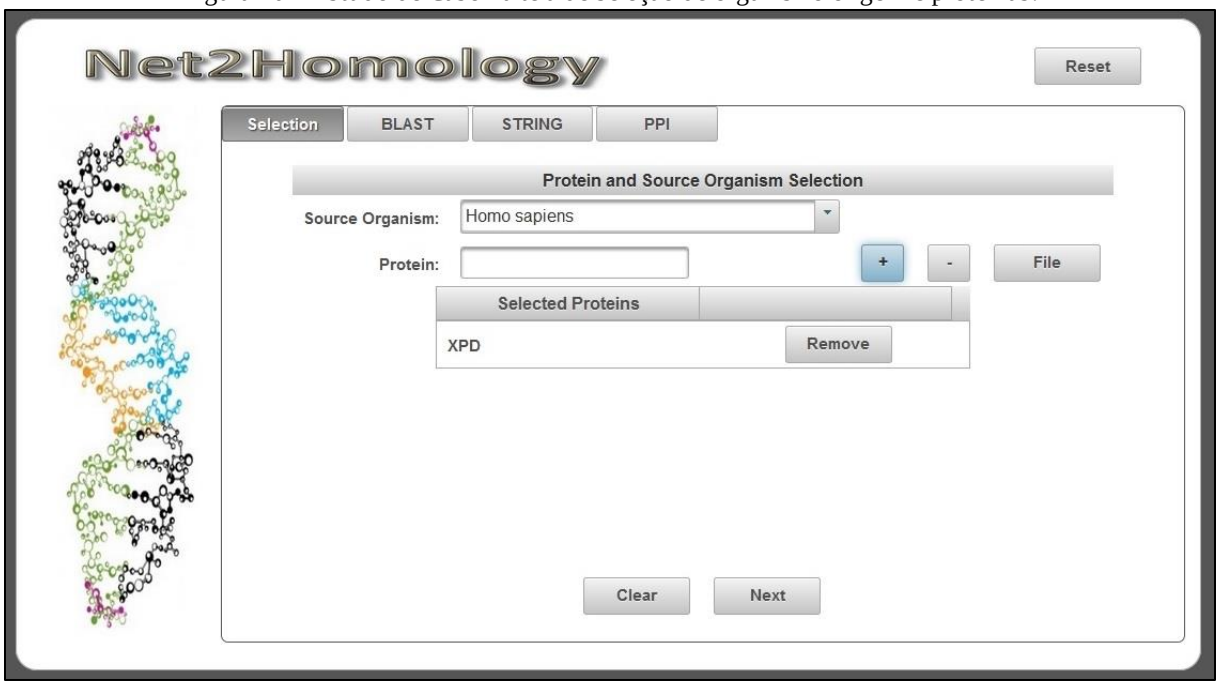
6 ESTUDO DE CASO

Nesse capítulo será apresentada uma execução completa da aplicação Net2Homology, já na plataforma *web*, apresentando também as funcionalidades desenvolvidas. Também serão realizadas algumas comparações da aplicação final com o que foi planejado durante a proposta de solução e também com a sua versão anterior, na plataforma *desktop*.

6.1 TELA DE SELEÇÃO DE ORGANISMO ORIGEM E PROTEÍNAS

A primeira etapa do *workflow*, assim como a primeira tela da aplicação, é a seleção do organismo origem e das proteínas que serão consultadas. Nessa etapa, selecionou-se o organismo *Homo sapiens* e a proteína XPD, conforme apresentado na Figura 20.

Figura 20 – Estudo de Caso na tela de seleção de organismo origem e proteínas.



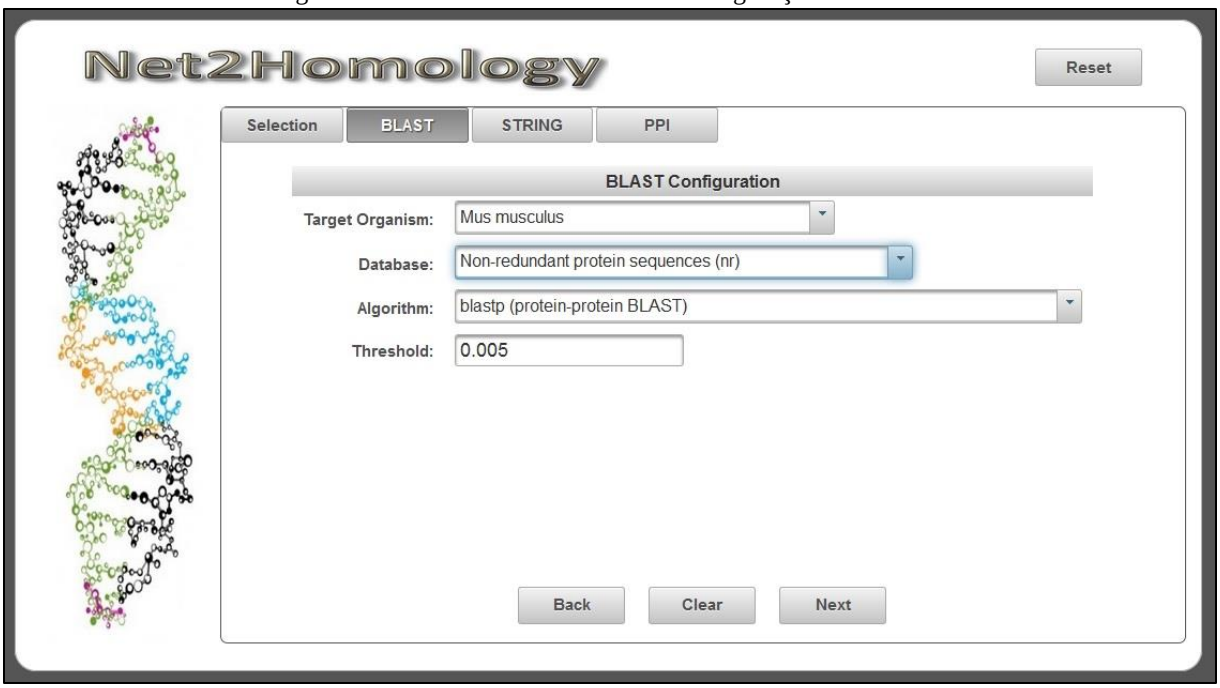
Fonte: autoria própria.

Diferente do que foi planejado, conforme seção 4.3.1 desse trabalho, não foi possível disponibilizar um campo para a realização de *upload* de arquivos de proteínas diretamente nessa tela, por restrições do *framework* utilizado. Para contornar essa situação, foi disponibilizado o botão *File*, conforme pode ser visualizado na Figura 20, que abre uma nova janela onde é possível informar um arquivo com as proteínas e então fazer o *upload* delas na aplicação. Também diferente da aplicação *desktop*, não é mais necessário selecionar as proteínas que se deseja consultar, após inseri-las na aplicação. Na versão *web*, todas as proteínas informadas na lista *Selected Proteins* serão consultas.

6.2 TELA DE CONFIGURAÇÃO DO BLAST

Após informar o organismo origem e as proteínas para consulta, é necessário informar o organismo alvo e configurar o BLAST informando o banco de dados, o algoritmo e o *e-value* máximo tolerável. Para a execução do estudo de caso, foi selecionado o organismo *Mus musculus*, o banco de dados *Non-redundant protein sequences (nr)*, o algoritmo *blastp (protein-protein BLAST)* e o *e-value* máximo 0.005, conforme pode ser observado na Figura 21.

Figura 21 - Estudo de caso na tela de configuração do BLAST



Fonte: autoria própria.

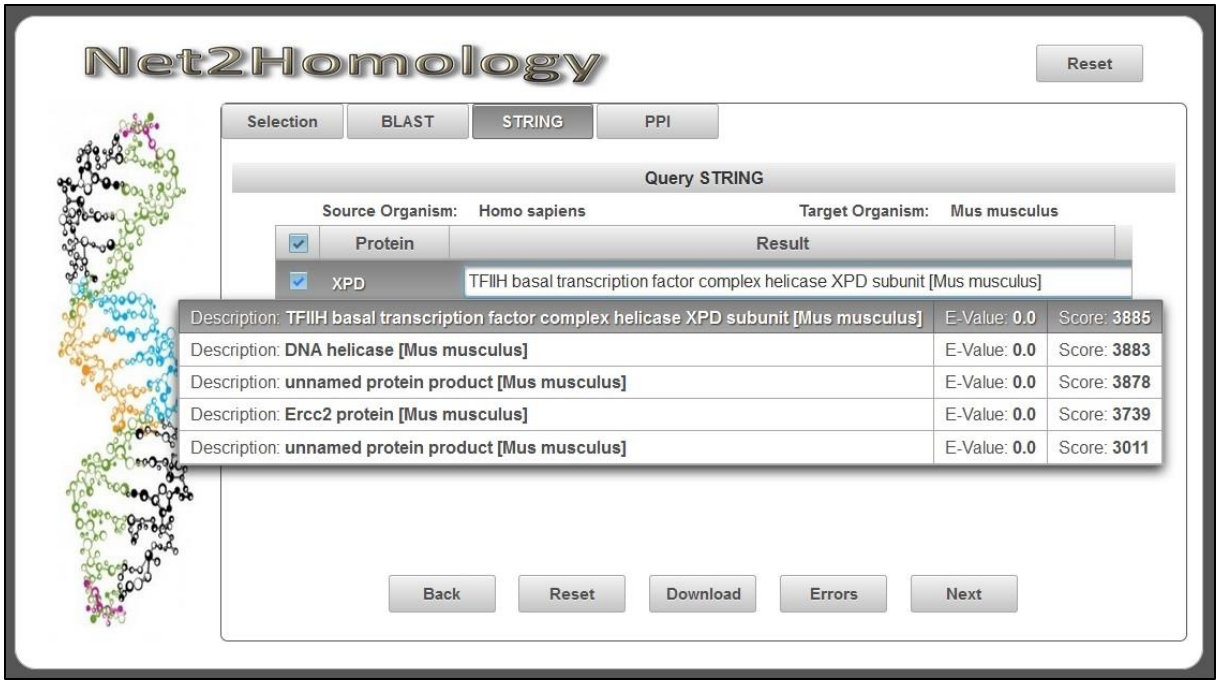
Diferente da aplicação *desktop*, não são mais necessárias duas etapas para a execução do BLAST e avanço à próxima etapa do *workflow*. Na nova aplicação, tudo ocorre ao clicar no botão *Next* da tela apresentada na Figura 21. Contudo, a execução do BLAST ocorre somente na primeira vez que o botão *Next* é clicado, mesmo que alterações na seleção de proteínas ou organismos sejam efetuadas. Essa tratativa foi realizada para evitar múltiplas execuções do BLAST, sempre que se clicasse no botão *Next*, mesmo sem ter realizado qualquer alteração nas seleções de proteínas ou organismos.

6.3 TELA DE RESULTADO DO BLAST E CONSULTA AO STRING

Após o término da execução do BLAST, é necessário realizar a seleção das proteínas da primeira etapa do *workflow* e dos resultados do BLAST recebidos para elas para realizar a consulta das redes PPI. Na execução do estudo de caso, foi selecionado o resultado do BLAST

da proteína XPE com o melhor resultado, que obteve uma pontuação de 3885 na comparação com a sequência original. A tela que apresenta essa seleção está ilustrada na Figura 22.

Figura 22 – Estudo de caso na tela de resultado do BLAST e consulta ao STRING.



Fonte: autoria própria.

Diferente do que foi planejado, conforme seção 4.3.3 deste trabalho, não são apresentados os campos de descrição da proteína, *e-value* e *score* devido à falta de espaço na tela. Para contornar essa situação, os campos de *e-value* e *score* são apresentados quando é aberta a lista para seleção do resultado, não prejudicando a análise para decidir qual resultado escolher. Quanto ao campo de descrição da proteína, a sua ausência não prejudicou o entendimento da tela sendo decidido pela sua retirada.

Também diferente do planejamento, o botão para *download* dos resultados não fica mais na mesma linha que o próprio resultado, assim como o botão para consulta das redes PPI. O botão de *download* foi trocado de lugar por conta da limitação do *framework* onde botões de *download* e *upload* não funcionam em formulários que recebem atualização por meio da execução de comandos em *ajax*. Como contorno à essa situação, o botão de *download* abre uma nova janela onde é possível selecionar o resultado para *download* conforme a proteína. Quanto ao botão para consultas das redes PPI, ele foi substituído pelo *check-box* de seleção dos registros, uma vez que a consulta das redes deve considerar múltiplas proteínas, sendo essa consulta realizada ao clicar no botão *Next* considerando todas os registros selecionados na lista.

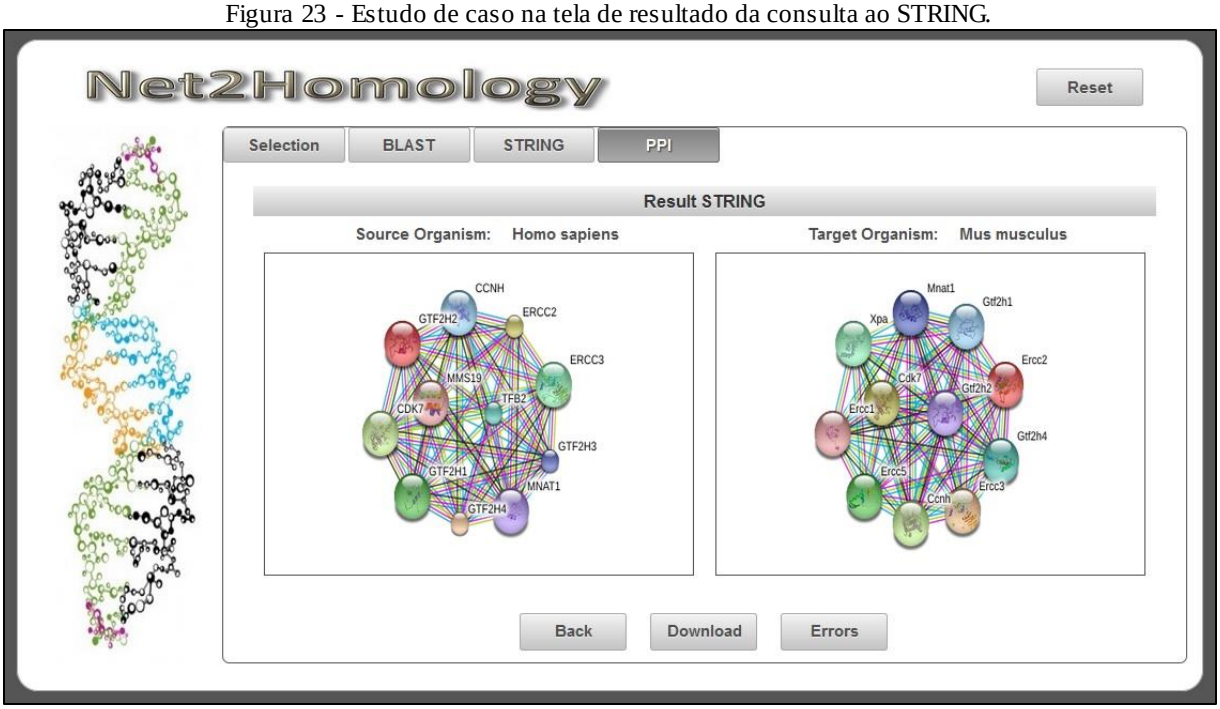
Assim como na execução do BLAST, essa tela também realiza a consulta das redes PPI apenas na primeira vez que o botão *Next* é clicado, mesmo que tenham sido feitas alterações

nas seleções dessa tela. Esse tratamento foi realizado dessa forma pelo mesmo motivo da execução do BLAST, evitar múltiplas consultas em bancos de dados remotos sem a realização de modificações nas seleções da tela, só que nesse caso é referente à consulta das redes PPI no banco de dados STRING.

6.4 TELA DE RESULTADO DA CONSULTA AO STRING

Ao término da consulta das redes PPI no banco de dados STRING, as imagens das redes PPI são apresentadas na última tela do *workflow* Net2Homology que foi abordada nesse trabalho. Em relação ao estudo de caso realizado e demonstrado, a tela para visualização das imagens das redes PPI está ilustrada na Figura 23.

Figura 23 - Estudo de caso na tela de resultado da consulta ao STRING.

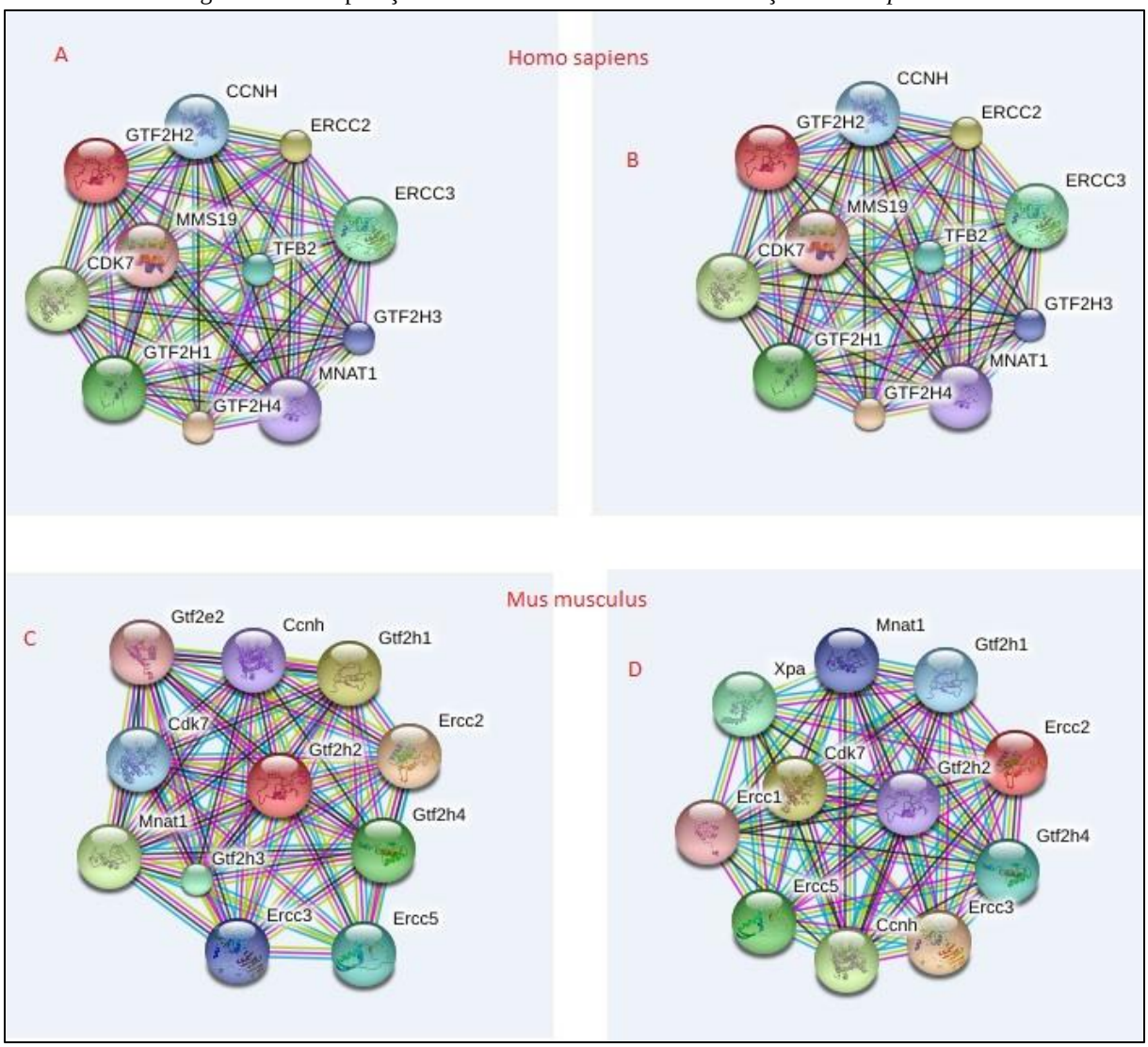


Fonte: autoria própria.

Diferente do que foi planejado, conforme relatado na seção 4.3.4, foram modificadas as opções de *download* que, por causa de limitações no *framework*, não permitem que botões com essa finalidade estejam no mesmo formulário onde campos são preenchidos através da execução de comandos em *ajax*. Para contorno dessa situação, o botão *Download*, apresentado na Figura 23, abre uma nova tela com os botões que permitem o *download* das informações das redes PPI apresentadas nessa etapa, tanto da sua imagem quanto dos seus dados. Também foi retirado o campo que apresentaria a proteína selecionada na etapa anterior por causa da possibilidade da seleção de múltiplas proteínas e, sendo assim, a presença do campo não melhoraria o entendimento da tela, sendo então decidido por sua retirada.

Comparando os resultados obtidos na execução da aplicação Net2Homology na plataforma *web*, com a execução, para os mesmos dados de entrada, na plataforma *desktop*, é possível observar que os resultados obtidos foram semelhantes, conforme apresentado na Figura 24, onde temos destacado em A e C, as redes PPI obtidas pela execução da ferramenta na plataforma *desktop*, e em B e D as redes PPI obtidas pela execução da ferramenta na plataforma *web*.

Figura 24 - Comparação dos resultados obtidos das execuções *desktop* e *web*.



Fonte: autoria própria.

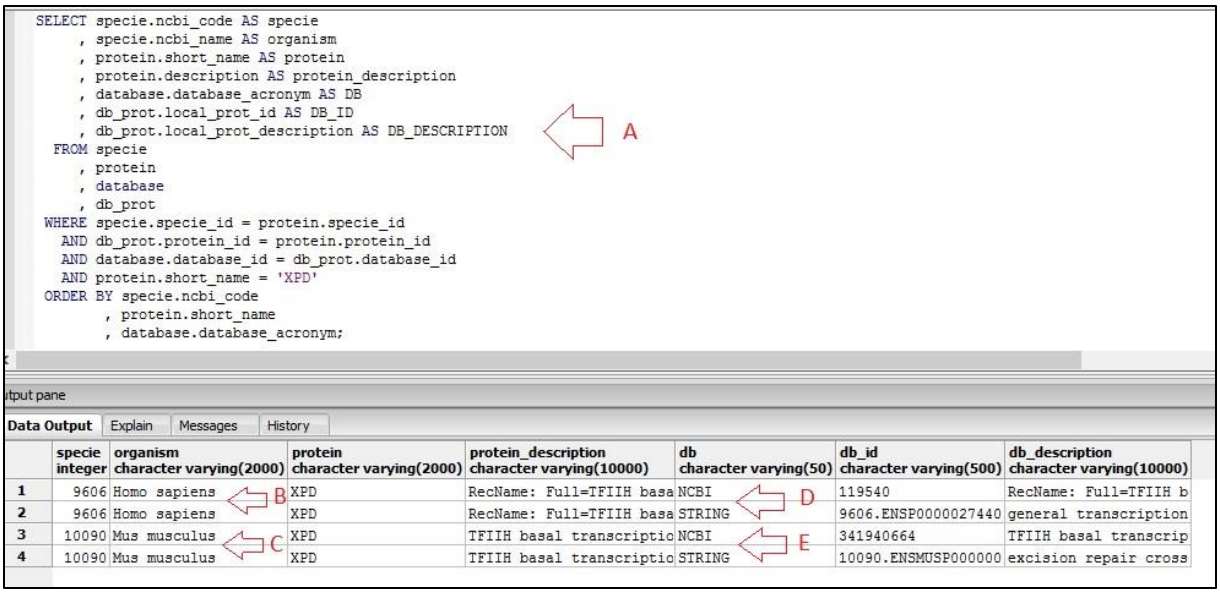
6.5 INTEGRAÇÃO NCBI X STRING

Uma das etapas de consulta das redes PPI no banco de dados STRING é a obtenção dos identificadores das proteínas selecionadas durante a execução do *workflow*. Esses identificadores provém do banco de dados STRING e são necessários para a consulta das redes PPI.

Nesse estudo de caso, a proteína XPD não existia no banco de dados local, por nunca ter sido consultada anteriormente. A avaliação da alteração do algoritmo de busca dos identificadores no banco de dados STRING foi feita mediante análise dos dados que constam no banco de dados local. Com base nisso, foi realizada uma consulta nesse banco de dados local, a fim de buscar os dados que o mesmo possui a respeito da proteína XPD.

Conforme ilustrado na Figura 25, foi realizada a consulta no banco de dados local, conforme destaque em A, buscando pela proteína XPD. Nessa consulta foram retornados quatro registros, dois referentes ao organismo *Homo sapiens*, conforme destaque em B e dois referentes ao organismo *Mus musculus*, conforme destaque em C. Cada um dos registros de um mesmo organismo está associado com um banco de dados diferente, sendo um deles vinculado com os bancos de dados do NCBI e o outro vinculado ao banco de dados STRING, como pode ser observado nos destaques em D e E.

Figura 25 - Verificação das referências das proteínas no banco de dados local.



```
SELECT specie.ncbi_code AS specie
     , specie.ncbi_name AS organism
     , protein.short_name AS protein
     , protein.description AS protein_description
     , database.database_acronym AS DB
     , db_prot.local_prot_id AS DB_ID
     , db_prot.local_prot_description AS DB_DESCRIPTION
  FROM specie
     , protein
     , database
     , db_prot
 WHERE specie.specie_id = protein.specie_id
   AND db_prot.protein_id = protein.protein_id
   AND database.database_id = db_prot.database_id
   AND protein.short_name = 'XPD'
 ORDER BY specie.ncbi_code
        , protein.short_name
        , database.database_acronym;
```

A

Data Output | Explain | Messages | History

	specie integer	organism character varying(2000)	protein character varying(2000)	protein_description character varying(10000)	db character varying(50)	db_id character varying(500)	db_description character varying(10000)
1	9606	Homo sapiens	XPD	RecName: Full=TFIIH basa	NCBI	119540	RecName: Full=TFIIH b
2	9606	Homo sapiens	XPD	RecName: Full=TFIIH basa	STRING	9606.ENSP0000027440	general transcription
3	10090	Mus musculus	XPD	TFIIH basal transcriptio	NCBI	341940664	TFIIH basal transcrip
4	10090	Mus musculus	XPD	TFIIH basal transcriptio	STRING	10090.ENSMUSP000000	excision repair cross

B C D E

Fonte: autoria própria.

Com base nos dados retornados da consulta ao banco de dados local, pode ser identificado que o algoritmo aprimorado está armazenando as relações entre as os bancos de dados do NCBI e o banco de dados STRING, quando encontra estas durante o processo de consulta dos identificadores. Dessa forma, quando consultada novamente essa mesma proteína, o seu identificador no banco de dados STRING será retornado consultando o banco de dados local, não sendo mais necessário fazer consultas em bancos de dados remotos para a obtenção dessa informação.

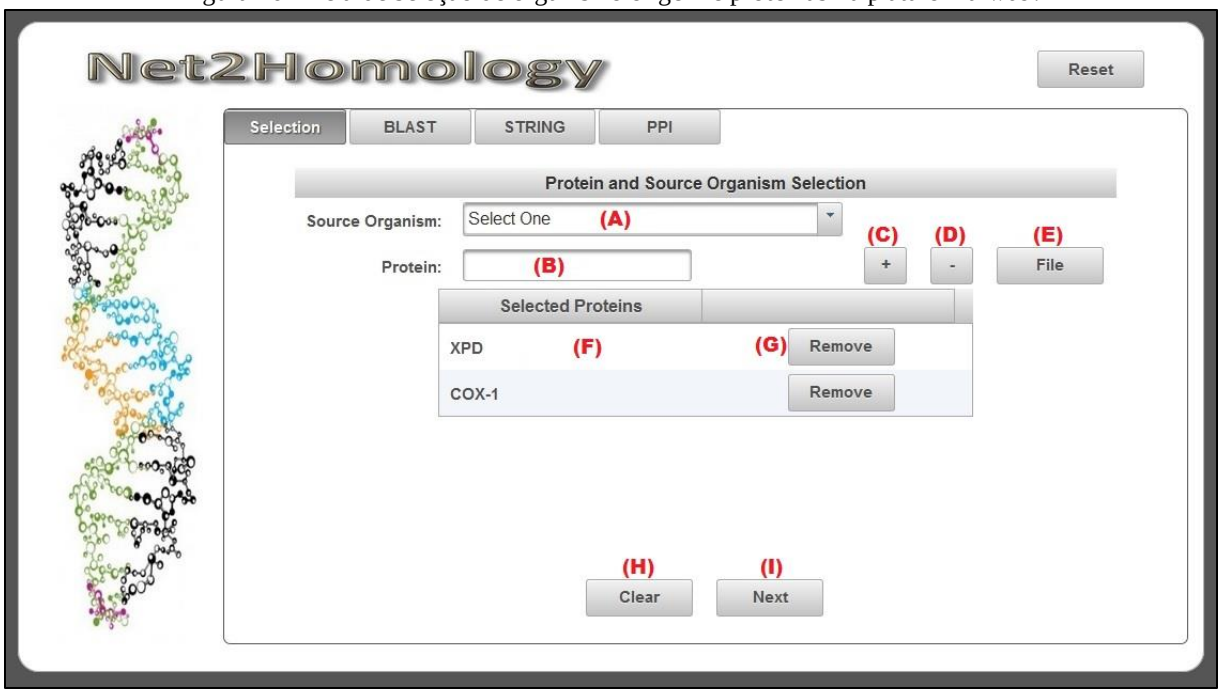
6.6 FUNCIONALIDADES DA APLICAÇÃO

Após verificar o estudo de caso, todas as funcionalidades desenvolvidas na aplicação Net2Homology na plataforma *web*, serão apresentadas, a fim de explicar como a utilização da mesma deve ocorrer. Para isso, serão apresentadas, tela por tela conforme a ordem de execução do *workflow*, seus campos e suas funções para a utilização da ferramenta.

6.6.1 Tela de Seleção de Organismo Origem e Proteínas

A tela de seleção de organismo origem e proteínas é a primeira tela da aplicação Net2Homology na plataforma *web*. Ela está ilustrada na Figura 26.

Figura 26 - Tela de seleção de organismo origem e proteínas na plataforma *web*.



Fonte: autoria própria.

O campo destacado em A na Figura 26 deve ser utilizado para informar o organismo origem, o qual será utilizado para busca das sequências fasta das proteínas selecionadas no *workflow*. Esse campo deve ser informado com uma das opções que o mesmo apresenta quando o mesmo recebe um clique do *mouse* com o botão direito. Para facilitar a consulta nessa lista, após abri-la clicando no campo, é possível digitar o nome ou parte do nome do organismo que a lista retorna apenas as opções que possuam o texto digitado.

Os campos destacados em B, C e D funcionam em conjunto e são utilizados para selecionar ou remover proteínas da aplicação. Quando informada uma proteína no campo destacado em B e clicado no botão destacado em C, a proteína é inserida no campo destacado

em F. Caso seja clicado no botão destacado em D, se a proteína existir em F, ela será removida. A fim de facilitar a remoção de proteínas da seleção, evitando a digitação de seu código, foi adicionado o botão destacado em G que remove a proteína à qual o botão está associado.

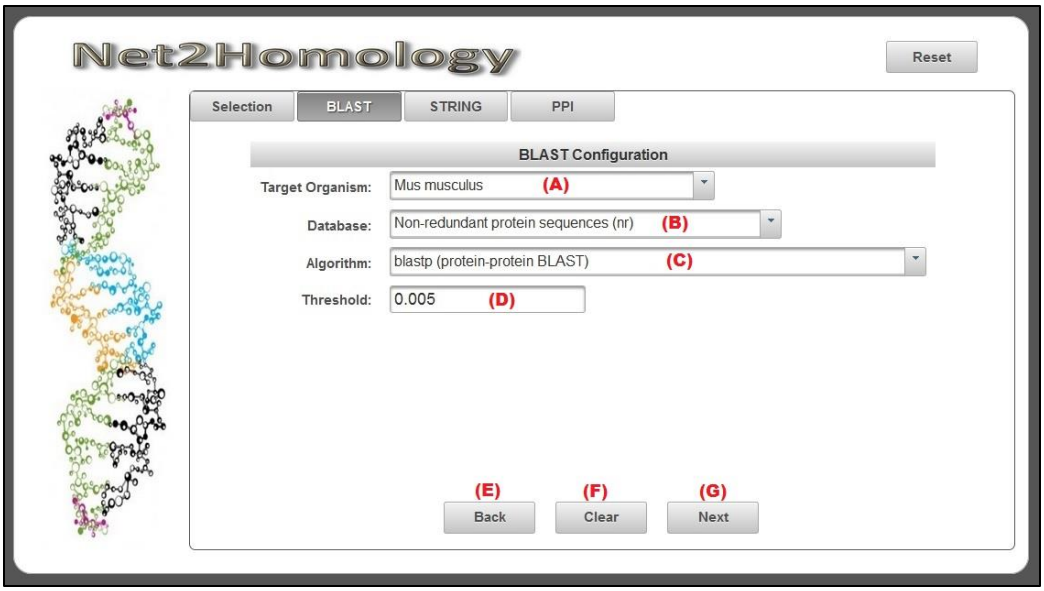
O botão destacado em E é um facilitador na seleção de proteínas. Quando clicado ele abre uma nova tela onde é possível selecionar um arquivo de proteínas, que esteja no formato TSV ou TSV-*no-header*, onde possua uma proteína na primeira coluna de cada linha e uma proteína por linha. Após selecionar um arquivo nesse formato, é possível realizar o *upload* do mesmo na aplicação onde as proteínas desse arquivo que não tenham sido selecionadas serão inseridas no campo destacado em F na Figura 26. Não há restrição na quantidade de arquivos selecionados, contudo só serão aceitos um arquivo por vez.

No botão destacado em H na Figura 26, quando utilizado, os campos destacados em A, B e F perderão qualquer valor que possuírem, reiniciando a tela para ser preenchida novamente. O campo destacado em I serve para prosseguir à próxima etapa do *workflow*. Nesse momento, serão realizadas as validações sobre os campos da tela, e só será possível prosseguir no *workflow* após informar um organismo origem e no mínimo uma proteína. Quando tentar prosseguir sem essas informações, são apresentadas mensagens para os campos serem preenchidos. Quando os campos forem informados, o *workflow* prosseguirá à próxima etapa.

6.6.2 Tela de Seleção do Organismo Alvo e Configuração do BLAST

Após passar pela primeira tela do *workflow*, será apresentada a tela de seleção do organismo alvo e configuração do BLAST. Essa tela está ilustrada na Figura 27.

Figura 27 - Tela de seleção do organismo alvo e configuração do BLAST.



Fonte: autoria própria.

No campo destacado em A na Figura 27, deve ser informado o organismo alvo, o qual será utilizado para a consulta pela execução do BLAST nos bancos de dados do NCBI. No campo destacado em B, deve ser informado em qual banco de dados do NCBI será realizado o BLAST. No campo destacado em C, deve ser informado qual o algoritmo de BLAST será utilizado na consulta. Esses três campos devem ser selecionados com uma das suas opções da lista que os mesmos apresentam quando são clicados. Para facilitar suas seleções, após abrir suas listas, é possível digitar um filtro, para reduzir a quantidade de opções apresentadas. Para os campos destacados em B e C, eles já vêm previamente preenchidos com valores iniciais.

O campo destacado em D deve ser informado com um número real, onde é aceito tanto a notação com vírgulas quanto a notação científica para calculadoras (exemplo 3E-5). Esse campo é a taxa de tolerância que a execução do BLAST considera durante a realização da consulta. Ele também possui um valor inicial. O botão destacado em E serve para retornar o *workflow* à etapa anterior. O botão destacado em F serve para limpar o campo destacado em A e para voltar os campos destacados em B, C e D aos seus valores iniciais.

O botão destacado em E, primeiramente, fará as validações de tela, verificando se todos os campos dessa tela foram informados. Também verifica se o organismo alvo dessa tela é diferente do organismo origem da tela anterior. Sob qualquer fato que não esteja conforme, mensagens são apresentadas informando dos problemas que devem ser corrigidos. Caso todos os dados da tela estejam conformes, esse botão inicia o processo de consulta nos bancos de dados do NCBI, por meio da execução do BLAST, abrindo uma linha de execução para cada proteína selecionada na etapa inicial do *workflow*. Com a conclusão de todas as linhas de execução, o *workflow* avança à próxima etapa. Caso esse botão seja clicado novamente, após já ter realizado o BLAST, ele não realizará nova consulta até que o *workflow* seja reiniciado, um processo que será explicado nas próximas seções.

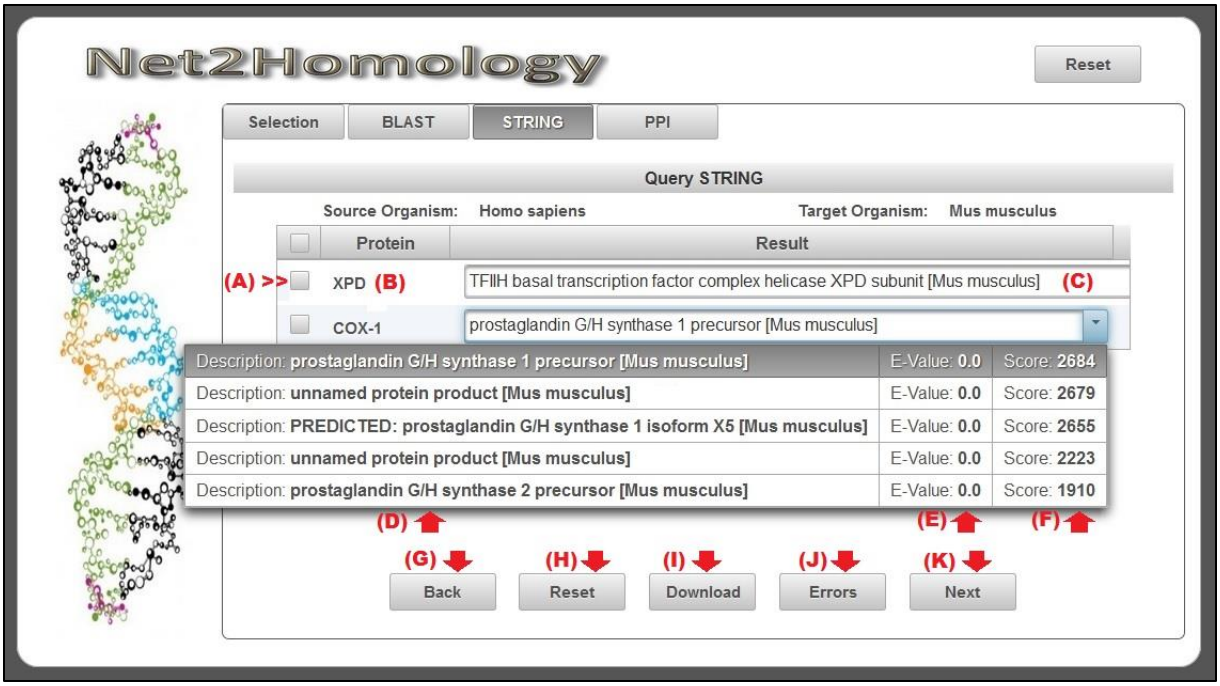
6.6.3 Tela de Resultados do BLAST e Consulta ao STRING

Com a conclusão da consulta nos bancos de dados do NCBI através da execução do BLAST, os resultados são processados e apresentados na tela da terceira etapa do *workflow* Net2Homology. Essa tela está ilustrada na Figura 28.

No campo destacado em A na Figura 28, está indicado uma caixa de checagem, que deve ser utilizada para selecionar a proteína e o resultado do BLAST retornado para ela na consulta da rede PPI no banco de dados STRING. No campo destacado em B, estão indicadas as proteínas da etapa inicial do *workflow* que tiveram resultados obtidos na execução do BLAST. Caso uma proteína não tenha obtido resultado, ela será apresentada na tela que é aberta

ao clicar no botão destacado em J juntamente com o erro que ocorreu e que impediu a obtenção do resultado do BLAST.

Figura 28 - Tela de resultado do BLAST e consulta ao STRING.



Fonte: autoria própria.

No campo destacado em C está uma lista de seleção do resultado do BLAST. Esse campo vem previamente preenchido com o melhor resultado obtido na execução do BLAST, avaliado segundo a pontuação, mas pode ser alterado por qualquer um dos até cinco resultados apresentados nesse campo. Quando clicado nesse campo, é aberta a lista que o mesmo possui, apresentando os campos destacados em D, E e F. O campo em D é a descrição do resultado obtido do relatório do BLAST, em E está o *e-value* do resultado e em F está a pontuação que o resultado obteve, calculada conforme a semelhança da sequência alvo com a sequência origem, durante a execução do BLAST.

O botão destacado em G na Figura 28 serve para retornar o *workflow* à etapa anterior. O botão destacado em H serve para desmarcar todos os resultados selecionados e fazer os campos de seleção dos resultados do BLAST voltarem aos valores iniciais, mostrando os melhores resultados obtidos. O campo destacado em I abre uma outra tela onde é possível realizar o *download* dos resultados do BLAST em formato XML, conforme a proteína. O campo destacado em J abre nova janela onde é possível visualizar as proteínas que não obtiveram resultado na execução do BLAST juntamente com o erro ocorrido.

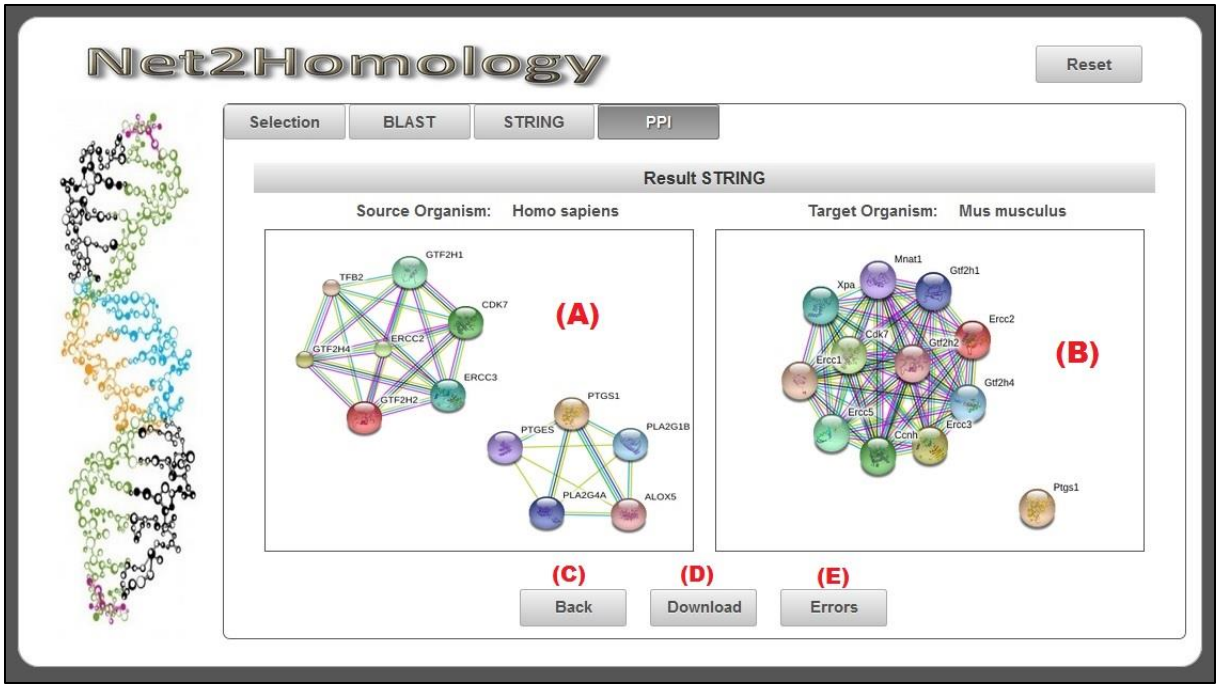
O campo destacado em K, primeiramente valida se algum resultado foi selecionado. Em caso negativo, é apresentada mensagem informando essa necessidade para prosseguir. Quando

houverem resultados selecionados, então o botão inicia a consulta das redes PPI, das proteínas e resultados selecionados nessa tela. Esse processo inicialmente busca os identificadores das proteínas no banco de dados STRING, conforme o algoritmo relatado na seção 4.3.5 desse trabalho. Com os identificadores das proteínas, então, é iniciada a consulta das redes de interação proteína-proteína no banco de dados STRING. Ao concluir essa consulta o *workflow* avança à próxima etapa. Caso esse botão seja clicado novamente, após já ter realizado a consulta das redes PPI, ele não irá realizar novas consultas até que o *workflow* seja reiniciado, um processo que será explicado nas próximas seções.

6.6.4 Tela de Visualização das Redes PPI

Ao concluir a consulta das redes PPI no banco de dados STRING, é apresentada a última tela do *workflow* que foi tratada nesse trabalho. A tela de visualização das redes PPI está ilustrada na Figura 29.

Figura 29 - Tela de visualização das redes PPI.



Fonte: autoria própria.

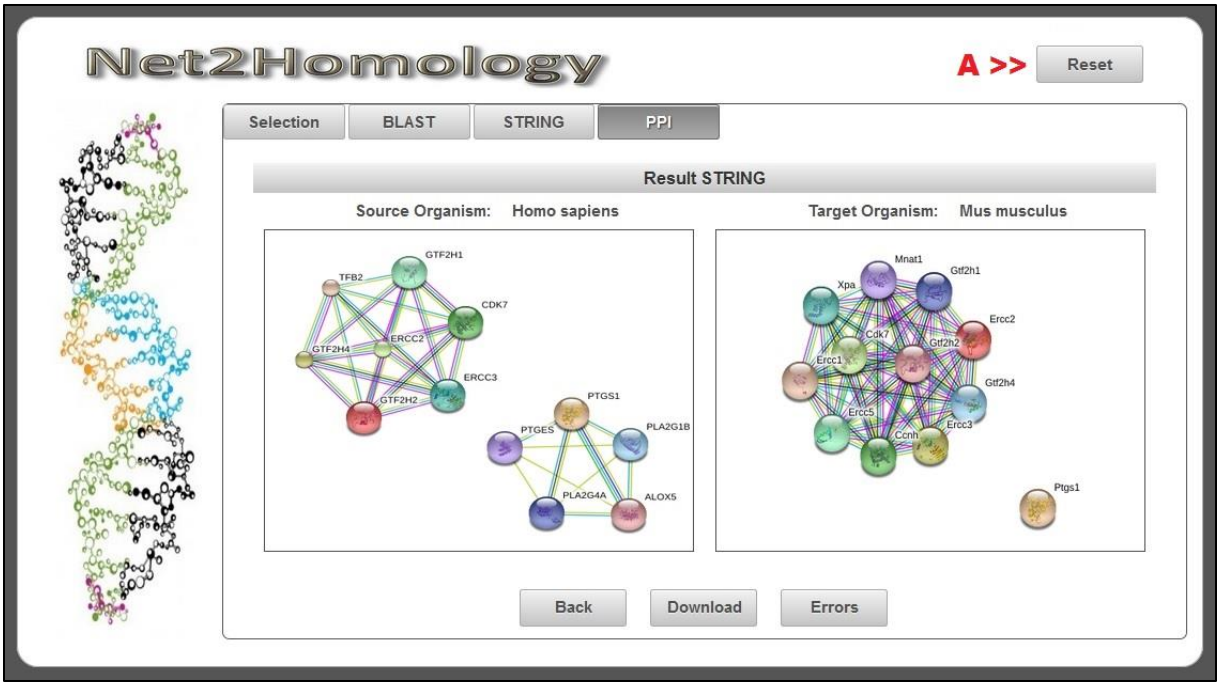
No campo destacado em A na Figura 29, é apresentada a imagem da rede PPI para o organismo origem, conforme a seleção realizada na etapa anterior. No campo destacado em B é apresentada a imagem da rede PPI para o organismo alvo, conforme seleção realizada na etapa anterior. O botão destacado em C serve para retornar o *workflow* à etapa anterior. O botão destacado em D abre nova tela onde é possível realizar o *download* tanto das imagens das redes PPI quanto dos seus dados em formato TXT. O botão destacado em E abre nova tela onde os

erros ocorridos durante o processo de consulta dos identificadores das proteínas no banco de dados STRING, ou mesmo durante o processo de consulta das redes PPI são apresentados.

6.6.5 Reiniciar o *Workflow*

Reiniciar o *workflow* é uma funcionalidade que foi mencionada nas seções 6.6.2 e 6.6.3. Ela foi desenvolvida porque os botões que fazem consultas em bancos de dados remotos, como na execução do BLAST e na consulta das redes PPI não as executam uma segunda vez, mesmo que os dados de seleção da aplicação sejam alterados. Desse modo, foi adicionado o botão destacado em A na tela ilustrada na Figura 30.

Figura 30 - Botão para reinício do *workflow*.



Fonte: autoria própria.

No botão destacado em A, que está disponível em qualquer uma das etapas do *workflow* Net2Homology na plataforma *web*, ao ser clicado irá limpar ou retornar aos valores iniciais todos os campos, em todas as telas do *workflow*. Após executar esse processo, nenhuma opção de *download* que estava disponível continuará disponível, assim como em nenhum campo os valores utilizados poderão ser recuperados. A funcionalidade desse botão se fez necessária para ser possível a execução do *workflow* múltiplas vezes de maneira consecutiva, o que, sem isso, não seria possível sem aguardar o servidor *web* expirar a sessão do navegador.

7 CONSIDERAÇÕES FINAIS

Finalizando esse trabalho com a análise dos resultados obtidos e apresentados durante o estudo de caso realizado nas seções 6.1 até 6.5 desse documento, conclui-se que os objetivos propostos foram atingidos quase em sua totalidade. A ferramenta Net2Homology foi aprimorada, executando agora na plataforma *web*. Referente a esse objetivo, só não foi completamente atingido porque não foi possível disponibilizar a ferramenta no servidor de hospedagem por causa de falhas que o mesmo apresentou. Esses problemas no servidor de hospedagem não foram solucionados até a conclusão desse trabalho. Contudo, a ferramenta foi executada completamente na plataforma *web*, ainda que em execuções locais, e a mesma se comportou da forma esperada, tanto no navegador Mozilla Firefox 33.1 quanto no Internet Explorer 11.0 e no Google Chrome 39.0.

O algoritmo de integração de dados entre os bancos de dados do NCBI e o banco de dados STRING também foi aprimorado, atingindo a esse objetivo também. Na versão *web* da ferramenta, os identificadores das proteínas são consultados no banco de dados STRING apenas quando os mesmos não existem no banco de dados local da aplicação. Nas situações onde os identificadores das proteínas não se encontram no banco de dados local, mas são encontrados durante a execução do algoritmo aprimorado, eles estão sendo armazenados localmente, criando as referências diretas entre os dados dos bancos de dados do NCBI e do banco de dados STRING.

Analisando as redes PPI obtidas como resultado da execução da aplicação na versão *web*, apresentadas na seção 6.4, essa nova integração fez com que o resultado dessa versão da ferramenta não fosse exatamente igual ao da versão *desktop*. Entretanto, a versão *desktop* apresentava falhas para algumas proteínas, como a COX-1, onde a rede PPI para o organismo *Homo sapiens* não era retornada porque não havia sido encontrado seu identificador no banco de dados STRING. Falha essa que, para a mesma proteína, a versão *web* da ferramenta não apresentou. Portanto, a alteração no algoritmo afetou a consulta da rede PPI, mas também melhorou o resultado obtido, onde foi possível encontrar identificadores de proteínas que na versão *desktop* não foi possível.

Ainda que os objetivos desse trabalho tenham sido atingidos quase em sua totalidade, algumas melhorias podem ser realizadas na aplicação Net2Homology, na sua versão *web*. Uma delas é realizar o trabalho de migração de plataforma, de *desktop* para a *web*, das etapas do *workflow* que não foram tratados no escopo desse trabalho. Uma das etapas que precisa ser migrada é a preparação dos dados e execução da ferramenta NetworkBlast. Outra etapa é a

apresentação da rede PPI recebida do resultado da execução do NetworkBlast, mostrando sua imagem e seus dados, disponibilizando-os para *download* e também visualizando-os em outras ferramentas como o Cytoscape Web.

Outra melhoria na aplicação é permitir ao usuário armazenar o estado do *workflow*, em qualquer ponto do mesmo, para ser recuperado posteriormente. Isso evitará que o usuário precise executar novamente as etapas do *workflow* que já foram executadas para um mesmo conjunto de dados e que, por algum motivo, não tenha sido possível concluir todas elas. Na versão *web* atual, se ocorrer algum problema que venha a abortar a execução do *workflow*, o usuário é obrigado a executar novamente todas as etapas, mesmo as que já teria executado anteriormente, por não conseguir salvar e recuperar o estado da aplicação. Essa melhoria também facilitaria a realização de múltiplas análises sobre um conjunto de dados onde, a cada execução, poucas alterações nesses dados seriam efetuadas.

8 REFERÊNCIAS

AMARAL, Alexandre Morais do; REIS, Marcelo da Silva; SILVA, Felipe Rodrigues da (Ed.). **O programa BLAST:** guia prático de utilização. Brasília: Embrapa, 2007. Disponível em: <http://ainfo.cnptia.embrapa.br/digital/bitstream/CENARGEN/29678/1/doc224.pdf>. Acesso em: 17 maio 2014.

BENSON, Dennis A. et al. GenBank. **Nucleic Acids Research: Database issue,** Oxford, v. 42, n. 1, p.D32-D37, jan. 2014. Disponível em: <http://nar.oxfordjournals.org/>. Acesso em: 26 abr. 2014.

BERNSTEIN, Philip A.. Middleware: A Model for Distributed System Services. **Communications Of The Acm,** New York, v. 39, n. 2, p.86-98, fev. 1996. Disponível em: <http://cacm.acm.org/magazines/1996/2/>. Acesso em: 19 maio 2014.

BOECKMANN, Brigitte et al. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. **Nucleic Acids Research,** Oxford, v. 31, n. 1, p.365-370, jan. 2003. Disponível em: <http://nar.oxfordjournals.org/>. Acesso em: 26 abr. 2014.

ELMASRI, Ramez; NAVATHE, Shamkant B.. **Sistemas de banco de dados.** 6. ed. São Paulo: Pearson, 2011. Tradução de: Daniel Vieira. Disponível em: <http://ucs.bv3.digitalpages.com.br/>. Acesso em: 21 abr. 2014.

FRANCESCHINI, Andrea et al. STRING v9.1: protein-protein interaction networks, with increased coverage and integration. **Nucleic Acids Reasearch: Database issue,** Oxford, v. 41, n. 1, p.D808-D815, jan. 2013. Disponível em: <http://nar.oxfordjournals.org>. Acesso em: 07 maio 2014.

GIBAS, Cynthia; JAMBECK, Per. **Desenvolvendo bioinformática.** Rio de Janeiro: Campus, 2001. 440 p.

GIL, Yolanda et al. Examining the Challenges of Scientific Workflows. **IEEE Computer,** Los Alamitos, v. 40, n. 12, p.24-32, dez. 2007. Disponível em: <http://eprints.soton.ac.uk/271187/1/computer07.pdf>. Acesso em: 18 maio 2014.

JENSEN, Lars J. et al. STRING 8—a global view on proteins and their functional interactions in 630 organisms. **Nucleic Acids Resear: Database issue,** Oxford, v. 37, n. 1, p.412-416, jan. 2009. Disponível em: <http://nar.oxfordjournals.org/>. Acesso em: 11 jun. 2014.

KARSCH-MIZRACHI, Ilene; NAKAMURA, Yasukazu; COCHRANE, Guy. The International Nucleotide Sequence Database Collaboration. **Nucleic Acids Research: Database issue,** Oxford, v. 40, n. 1, p.D33-D37, jan. 2012. Disponível em: <http://nar.oxfordjournals.org/>. Acesso em: 01 jun. 2014.

KÖHLER, Jacob; PHILIPPI, Stephan; LANGE, Matthias. SEMEDA: ontology based semantic integration of biological databases. **Bioinformatics,** Oxford, v. 19, n. 18, p.2420-2427, jun. 2003. Disponível em: <http://bioinformatics.oxfordjournals.org>. Acesso em: 25 maio 2014.

LESK, Arthur M.. **Introdução à Bioinformática.** 2. ed. Porto Alegre: Artmed, 2008. Tradução de: Ardala Elisa Breda Andrada, et al.

LU, Hongchao et al. The interactome as a tree—an attempt to visualize the protein–protein interaction network in yeast. **Nucleic Acids Research,** Oxford, v. 32, n. 16, p.4804-4811, set. 2004. Disponível em: <http://nar.oxfordjournals.org/>. Acesso em: 27 jun. 2014.

LUSHBOUGH, Caro M.; GNIMPIEBA, Etienne; DOOLEY, Rion. BioExtract Server, a Web-based Workflow Enabling System, Leveraging iPlant Collaborative Resources. **2013 Ieee International Conference On Cluster Computing (cluster),** Los Alamitos, p.1-3, set. 2013. DOI: 10.1109/CLUSTER.2013.6702692. Disponível em: <http://ieeexplore.ieee.org/>. Acesso em: 19 jun. 2014.

NCBI (Eua). National Institutes Of Health. Database resources of the National Center for Biotechnology Information. **Nucleic Acids Research: Database issue,** Oxford, v. 42, n. 1, p.7-17, jan. 2014. Disponível em: <http://nar.oxfordjournals.org/>. Acesso em: 08 jun. 2014.

NOTARI, Daniel Luis. **Desenvolvimento de workflows científicos para a geração e análise de diferentes redes de interatomas**. 2012. 180 f. Tese (Doutorado em Biotecnologia) – Programa de Pós-Graduação em Biotecnologia, Universidade de Caxias do Sul, Caxias do Sul. 2012.

PUGA, Sandra; FRANÇA, Edson; GOYA, Milton. **Banco de Dados:** Implementação em SQL PL/SQL e Oracle 11g. São Paulo: Pearson, 2014. Disponível em: <http://ucs.bv3.digitalpages.com.br/>. Acesso em: 21 abr. 2014.

SILVA, Fabrício Nogueira da; CAVALCANTI, Maria Cláudia; DÁVILA, Alberto M. R.. In Services: Data Management for In Silico Workflows. **Dexa '06. 17th International Workshop On Database And Expert Systems Applications, 2006.,** Los Alamitos, p.206-210, set. 2006. DOI: 10.1109/DEXA.2006.72. Disponível em: <http://ieeexplore.ieee.org/>. Acesso em: 19 jun. 2014.

SZKLARCZYK, Damian et al. The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. **Nucleic Acids Research: Database issue,** Oxford, v. 39, n. 1, p.D561-D568, jan. 2011. Disponível em: <http://nar.oxfordjournals.org/>. Acesso em: 22 abr. 2014.