UNIVERSIDADE DE CAXIAS DO SUL
Centro de Computação e Tecnologia da Informação
Curso de Bacharelado em Ciência da Computação

Cristian Reolon

CONSULTA A REDES DE INTERAÇÕES DE PROTEÍNAS A PARTIR DA EXTRAÇÃO DE DADOS BIOLÓGICOS

Caxias do Sul

2011

Cristian Reolon

CONSULTA A REDES DE INTERAÇÕES DE PROTEÍNAS A PARTIR DA EXTRAÇÃO DE DADOS BIOLÓGICOS

Trabalho de Conclusão de Curso para obtenção do Grau de Bacharel em Ciência da Computação da Universidade de Caxias do Sul.

Prof. MSc. Daniel Luís Notari
Orientador

Caxias do Sul

2011

“Para obter algo que nunca teve,
precisa fazer algo que nunca
fez”.
Chico Xavier

AGRADECIMENTOS

Agradeço aos meus pais, Orilde e Luiz, pelo amor, carinho e incentivo oferecido durante todos os anos de faculdade e de vida. A minha família que sempre esteve do meu lado, colaborando de alguma forma para o meu crescimento.

A minha noiva Vanessa, meus mais sinceros agradecimentos pelo amor, amizade, dedicação e compreensão nos diversos momentos que não estive presente por necessitar concluir os trabalhos curriculares.

Aos meus amigos e colegas que sempre estavam dispostos a ajudar e contribuir de alguma forma para o meu aprendizado.

Aos meus professores, pela paciência, dedicação e aptidão para difundir o conhecimento e formar opiniões também demonstro minhas sinceras considerações. Em especial ao professor Daniel Luis Notari e a professora Helena Graziottin Ribeiro que sempre estavam dispostos a conversar e contribuir para a conclusão deste trabalho.

E por fim, agradeço a Deus por todas as bênçãos que tive na vida, proporcionando caminhos para todas as conquistas da minha vida.

SUMÁRIO

LISTA DE ABREVIATURAS E SIGLAS

Sigla	Significado em Português	Significado em Inglês
BLAST	Ferramenta de Busca de Alinhamento Local	*Basic Local Alignment Search Tool*
DNA	Ácido Desoxirribonucleico	*Deoxyribonucleic Acid*
NCBI	Centro Nacional de Informações sobre Biotecnologia	*National Center for Biotechnology Information*
NIH	Instituto Nacional de Saúde	*National Institutes of Health*
OMIM		*Online Mendelian Inheritance in Man*
PATHBLAST	Ferramenta para alinhamento de redes de proteínas	
PHP		*PHP: Hypertext Preprocessor*
RNA	Ácido Ribonucleico	*Ribonucleic Acid*
STRING	Ferramenta de busca para recuperação de interação de genes e proteínas	*Search Tool for the Retrieval of Interacting Gene/Proteins*
URL	Localizador Universal de Recursos	Universal Resource Locator
XML	Linguagem de Modelagem Extensível	Extensive Markup Language

LISTA DE FIGURAS

RESUMO

Este trabalho apresenta um estudo sobre integração de dados utilizando os serviços disponibilizados pelos bancos de dados biológicos, OMIM, STRING e PathBLAST. Oldra (2009), em um trabalho prévio desenvolveu um sistema *web* para integrar os *sites* OMIM e STRING com exceção da consulta ao *site* do PathBLAST. O presente trabalho de conclusão visou desenvolver um sistema similar ao desenvolvido por Oldra (2009) na linguagem de programação Java com novas funcionalidades, dentre elas, a consulta ao *site* do PathBLAST possibilitando ao usuário consultar doenças genéticas, automatizando e integrando os *sites* ao software utilizado pelo especialista.

Palavras-chave: Bioinformática, Doenças Genéticas, Integração de Dados, Processo Biológico.

1 INTRODUÇÃO

Iniciado oficialmente em 1990, o Projeto Genoma Humano é um programa coordenado pelo U.S. Department of Energy (U.S. Department of Energy, 2004) e National Institutes of Health (NIH, 2004). As principais diretrizes do projeto (HGP, 2004b):

- Mapear e sequenciar o genoma humano inteiro, obtendo as 3 bilhões de bases da cadeia que representa o DNA humano;
- Identificar os aproximadamente 30.000 genes no DNA humano;
- Armazenar esta informação em bancos de dados;
- Melhorar as ferramentas de análises dos dados;
- Transferir as tecnologias relacionadas ao setor privado; e
- Tratar das conseqüências éticas, legais e sociais que surgirem com o projeto.

Os projetos para estudo de genomas partem de uma fase de sequenciamento onde são gerados em laboratórios dados brutos, ou seja, sequências de DNA sem significado biológico. As sequências de DNA possuem códigos responsáveis pela produção de proteínas e RNAs, enquanto que as proteínas participam de todos os fenômenos biológicos, como a replicação celular, produção de energia, defesa imunológica, contração muscular, atividade neurológica e reprodução (LEMOS, 2004).

Durante a fase de análise das seqüências de DNA, RNA e proteínas, os pesquisadores usam diversas ferramentas, programas de computador, e um grande volume de informações armazenadas em fontes de dados de Biologia Molecular. O crescente volume, a distribuição das fontes de dados e a implementação de novos processos em Bioinformática facilitaram enormemente a fase de análise, porém criaram uma demanda por ferramentas e sistemas semi-automáticos para lidar com tal volume e complexidade (LEMOS, 2004).

As redes são basicamente descritas como conjuntos de itens conectados entre si e são observadas em inúmeras situações, desde o nível subatômico até as mais complicadas estruturas sociais ou matérias concebidas pela humanidade. Em grande parte das vezes,

verifica-se que o estudo dos elementos que compõe a rede é insuficiente para explicar o seu comportamento observável. A importância do estudo das redes vem da sua identificação nas mais diversas situações, como em sistemas químicos, orgânicos e sociais (NEWMAN, 2003).

A enorme quantidade de dados disponíveis sobre a interação de redes de proteínas levantaram novas questões sobre a evolução da rede e funções. Esses dados também introduzem um grande número de desafios técnicos: como separar a proteína-proteína e proteínas-interação DNA dos falsos positivos. Como anotar as interações com regras funcionais e, em última instância, como organizar os dados de interação de larga escala em modelos de sinalização celular e máquinas de regulação. Como é no caso na biologia, uma abordagem baseada no cruzamento de espécies, a comparação pode fornecer um quadro valioso para abordar esses desafios. Assim como o BLAST é usado para executar o alinhamento da seqüência de proteína/DNA, o PathBLAST baseia-se no alinhamento da rede de proteínas (NCBI, 2004).

Uma rede (grafo) é uma coleção de pontos onde esses pontos são chamados de nodos ou vértices, e os arcos que conectam estes pontos são chamados de arestas. Redes biológicas, representações de relacionamentos biológicos, são construídas para descrever vários fenômenos biológicos. Estas redes variam desde redes que descrevem condutores bioquímicos da célula até redes de mais alto nível tais como redes de neurônios (BEBEK. YANG, 2007).

Especificamente, o PathBLAST procura pelo alinhamento do caminho de máxima pontuação entre dois caminhos, um para cada rede, em que proteínas do primeiro caminho são emparelhadas com ocorrências ortólogos putativas na mesma ordem do segundo caminho.

O alinhamento do caminho é marcado pelo grau de similaridade da seqüência de proteínas em cada posição do caminho e pela qualidade da interação da proteína que contêm. Para lidar com os erros experimentais e a variação evolucionária entre as redes, o método também permite falhas no alinhamento do caminho. A falha ocorre quando as proteínas que interagem em um caminho são alinhadas contra proteínas ortólogas em outro caminho que não interagem diretamente, mas são ligas a duas distâncias (ou seja, ambos interagem através de uma proteína em comum). PathBLAST implementa uma pesquisa através de todos

alinhamentos possíveis entre duas redes diferentes para identificar os alinhamentos de maior pontuação em geral (PathBLAST, 2005).

1.1 Motivação

Este trabalho visa criar meios que facilitem o trabalho dos profissionais de biologia na consulta e processamento de dados ligados aos bancos de dados biológicos OMIM, STRING e PathBLAST.

O que motiva a realização do trabalho é poder disponibilizar ferramentas que sejam úteis ao trabalho diário dos profissionais de biologia e que possibilite centralizar as pesquisas em um único local.

1.2 Problema de Pesquisa

Com base no trabalho desenvolvido por Oldra (2009), a entrada de dados será realizada através do sistema desenvolvido na linguagem de programação Java, permitindo que sejam pesquisadas redes de interação de proteínas sem a necessidade do acesso manual às páginas *Web* OMIM e STRING. A nova funcionalidade apresentada efetuará a consulta também ao *site* do PathBLAST buscando o alinhamento da rede de proteínas. O sistema se encarregará de fazer a comunicação com os sites, conterá melhorias de usabilidade e permitirá detectar o relacionamento entre as seqüências que possuem similaridade.

1.3 Objetivos

Os objetivos do trabalho são os seguintes:

1. O objetivo geral deste trabalho é desenvolver um sistema de consulta aos bancos de dados biológicos possibilitando a realização de análises através do processamento das redes de interações de proteínas.

2. Estudar o problema da consulta a redes de proteínas e bancos de dados biológicos e desenvolver uma implementação que permita complementar o sistema desenvolvido em um

trabalho prévio (OLDRA, 2009), criando mais formas de consultas utilizando a linguagem de programação Java.

1.4 Estrutura do trabalho

Este trabalho está organizado da seguinte forma:

O capítulo 2 apresenta os conceitos pertinentes à bioinformática e automação de processos.

O capítulo 3 apresenta a proposta de software para atender a necessidade levantada com os profissionais da área da biologia.

O capítulo 4 apresenta o desenvolvimento do sistema de consulta a redes de interação de proteínas a partir da extração de bancos de dados biológicos explicando a implementação realizada.

O capítulo 5 descreve conclusões baseadas no desenvolvimento deste trabalho e nos resultados obtidos, bem como sugestões de trabalhos futuros.

2 BIOINFORMÁTICA

Nesse capítulo serão apresentados os conceitos de bioinformática, biologia molecular e bancos de dados biológicos.

2.1 Bioinformática

A bioinformática é um subconjunto de um campo maior da biologia computacional, a aplicação de técnicas analíticas quantitativas à modelagem de sistemas biológicos (GIBAS; JAMBECK, 2001).

A bioinformática surgiu da necessidade de se compreender as funções biológicas, é responsável por armazenar e relacionar dados biológicos com o auxílio de algoritmos matemáticos e métodos computacionais capazes de analisar grande quantidade de dados biológicos. Possui relação com a engenharia de software, matemática, física, química, estatística, ciência da computação e biologia molecular (BRASIL ESCOLA - BIOINFORMÁTICA, 2008).

2.2 Biologia Molecular

O campo de estudo da biologia molecular são as interações bioquímicas celulares envolvidas na duplicação do material genético e na síntese protéica. Consiste principalmente em estudar as interações entre os vários sistemas de células, partindo da relação entre o DNA, o RNA e a síntese de proteínas, e o modo como essas interações são reguladas (BIOLOGIA MOLECULAR, 2004).

O DNA contém toda a informação genética dos indivíduos, uma molécula formada por duas cadeias em forma de dupla hélice ligadas entre si onde seu principal papel no organismo é de armazenar as informações necessárias para a síntese e replicação de proteínas. O RNA

participa do processo de formação de proteínas a partir da transcrição da molécula de DNA (ALGO SOBRE, 2000).

Proteínas são corpos definidos por sequencias de aminoácidos. Para cada proteína existe um segmento do DNA que guarda informações sobre ela, detalhando sua sequência de aminoácidos, sendo que a forma e outras propriedades de cada proteína são dadas pela sequência de aminoácidos que a constitui (Raychaudhuri, 2006).

Em sua estrutura primária cada proteína é representada por uma sequência de aminoácidos (seqüência de letras). O objetivo de alinhar proteínas é estabelecer parâmetros de semelhança entre duas ou mais proteínas, respeitando critérios específicos e respeitando a sequencialidade dos átomos.

Uma maneira de fazer a comparação de sequências é computar um alinhamento entre as sequências correspondentes às moléculas de interesse (ou fragmentos dessas moléculas). Um alinhamento é uma maneira de inserir espaços nas sequências de modo que todas fiquem com mesmo comprimento, para possibilitar uma fácil comparação.

A comparação entre sequências de DNA de organismos diferentes é baseada no conceito de que estes organismos originam-se de um ancestral em comum (BRITO, 2003).

Uma rede (grafo) é uma coleção de pontos onde estes pontos são chamados de nodos ou vértices, e os arcos que conectam estes pontos são chamados de arestas. Redes biológicas, representações de relacionamentos biológicos, são construídas para descrever vários fenômenos biológicos. Estas redes variam desde redes que descrevem condutores bioquímicos da célula até redes de mais alto nível tais como redes de neurônios (BEBEK; YANG, 2007).

Segundo (MedicineNet, 2011) a doença genética é qualquer doença causada por uma anomalia no genoma de um indivíduo. A anomalia pode variar a partir de uma pequena mutação em uma única base no DNA de um único gene para uma anomalia cromossômica grave envolvendo a adição ou subtração de um cromossomo inteiro ou um conjunto de cromossomos. Algumas doenças genéticas são herdadas dos pais, enquanto outras são causadas por alterações adquiridas ou mutações em um gene pré-existente ou grupo de genes.

2.3 Workflow científico

Um *workflow* diz respeito à automação de procedimentos, onde documentos, informações ou tarefas são passadas entre os participantes de acordo com um conjunto pré-definido de regras, para se alcançar ou contribuir no objetivo global de um negócio. Apesar de um workflow permitir a organização manual, na prática a maioria dos workflows são organizados dentro de um contexto do sistema de informação para prover um apoio automatizado aos procedimentos (WfMC, 2004).

Workflows científicos são definidos como resolução de problemas científicos através de técnicas tradicionais de workflows. Ou seja, as idéias de execução de um conjunto de tarefas de uma determinada sequência foram aproveitadas na área científica para a realização de experimentos e estudos. Nos workflows científicos os passos são na maioria das vezes compostos por programas computacionais que recebem, processam e geram um conjunto de dados científicos que podem ser repassados aos demais passos do workflow (SILVA, 2006).

2.4 Bancos de dados biológicos

Um banco de dados pode ser definido como uma coleção compartilhada de dados logicamente relacionados, projetado para atender as necessidades de informação de múltiplos usuários em uma organização. Os bancos de dados armazenam dados relativos a um domínio particular e representam algum aspecto do mundo real, o qual deve ser mantido consistente dentro do banco de dados.

Um banco de dados biológico constitui um grande conjunto de dados persistentes, geralmente associados a um software projetado para atualizar, consultar e recuperar componentes de dados armazenados no sistema (ROCHA, 2007).

Bancos de dados biológicos são, geralmente, tabelas que possuem grandes quantidades de registros. E os registros podem ser da seguinte maneira: um registro associado a uma sequência de proteínas contém normalmente uma descrição do tipo de molécula, seu nome científico e citações na literatura que correspondem a esta sequência.

2.4.1 OMIM

OMIM[1] (*Online Mendelian Inheritance in Man*) é uma base de dados que contêm um catálogo de genes humanos e doenças genéticas e foi desenvolvida pelo NCBI (*National Center for Biotechnology Information*) (OMIM Help Document, 2005).

OMIM disponibiliza três maneiras de pesquisar doenças genéticas ou informações relacionadas:

1) Pesquisa normal: digitando uma palavra-chave, como no caso da maioria dos bancos de dados.
2) Mapa de genes: procura uma tabela de genes organizados pelo mapa de localização citogenética.
3) Mapa mórbido: tabela de todas as doenças listadas em ordem alfabética genético.

Segue abaixo a sequência de passos para efetuar a pesquisa de uma doença:

1) O processo de pesquisa de uma doença começa acessando o site do OMIM.

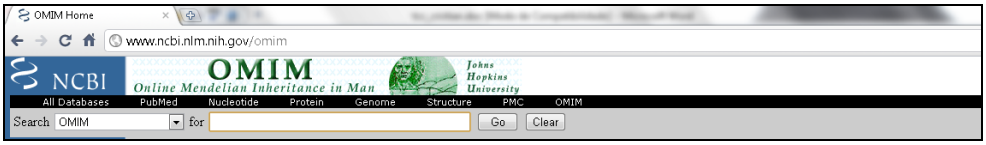

Figura 1 – Acesso ao site do OMIM (OMIM, 2005)

2) No campo *Search* selecionamos a opção OMIM, no campo *for* digitamos o nome da doença em inglês e clicamos no botão *Go*.

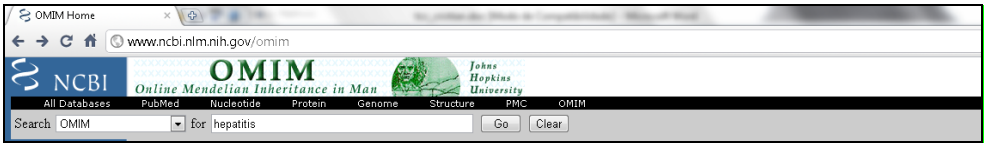

Figura 2 – Pesquisa da doença (OMIM, 2005)

[1] OMIM Online Mendelian Inheritance in Man. Disponível em: <http://www.ncbi.nlm.nih.gov/omim>. Acesso em: 29 de março de 2011

3) O site apresentará uma lista com as ocorrências da doença para selecionar a que você está procurando, conforme mostra a Figura 3.

Figura 3 – Lista de ocorrências da doença (OMIM, 2005)

2.4.2 STRING

STRING[2] (*Search Tool for the Retrieval of Interacting Gene/Proteins*) é um banco de dados de interações de proteínas conhecidas e previsíveis. As interações incluem associações diretas (físicas) e indiretas (funcionais) e são provenientes de quatro fontes:

- Contexto genômico
- Experimentos de alta capacidade
- Coexpressões conservadas
- Conhecimento prévio

STRING integra dados de interação dessas fontes para um grande número de organismos, e as transferências de informações entre estes organismos, quando aplicável. O banco de dados abrange atualmente 2.590.259 proteínas de 630 organismos.

[2] STRING. Disponível em: <http://string.embl.de>. Acesso em: 29 de março de 2011.

1) O processo de pesquisa de uma proteína começa acessando o site do STRING.

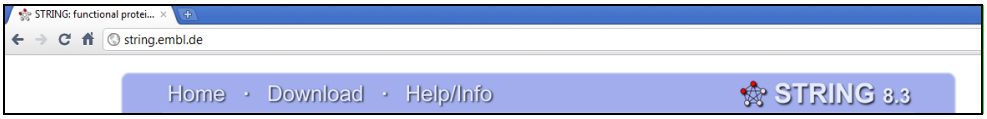

Figura 4 – Acesso ao site do STRING (STRING, 2000)

2) No campo *protein name* digitamos o nome da proteína e clicamos no botão *Go.*

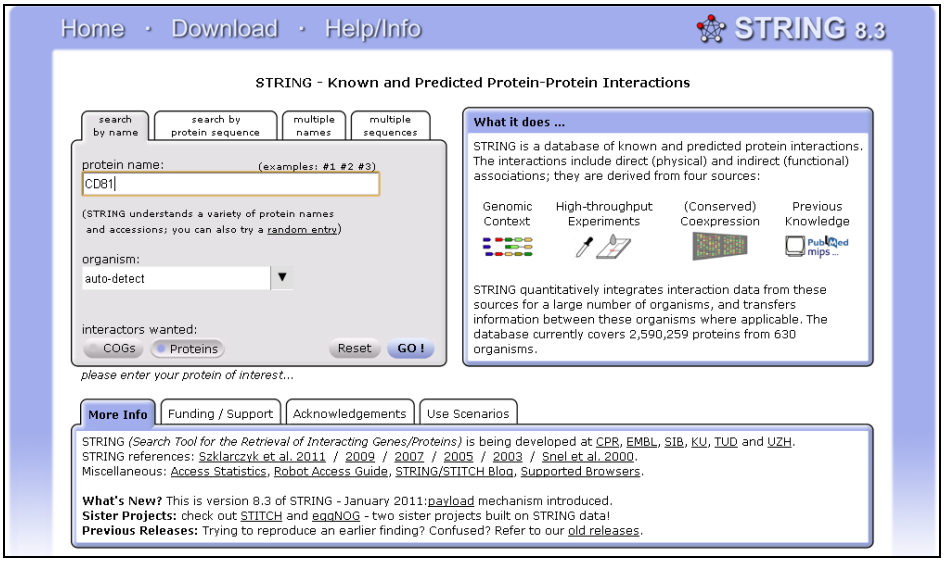

Figura 5 – Pesquisa da proteína (STRING, 2000)

3) O site apresentará uma lista de organismos para selecionar, como mostra a Figura 6. Selecione o organismo que desejar e clique no botão *Continue.*

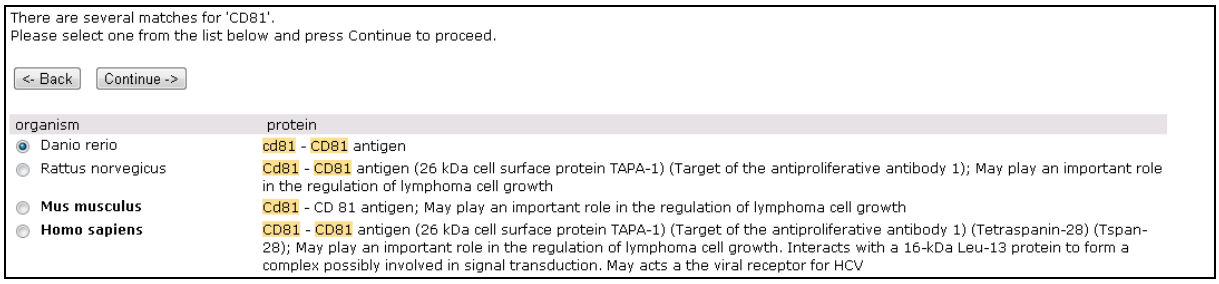

Figura 6 – Lista de ocorrências da proteína (STRING, 2000)

4) Será apresentada a rede de interação da proteína, conforme mostra a Figura 7.

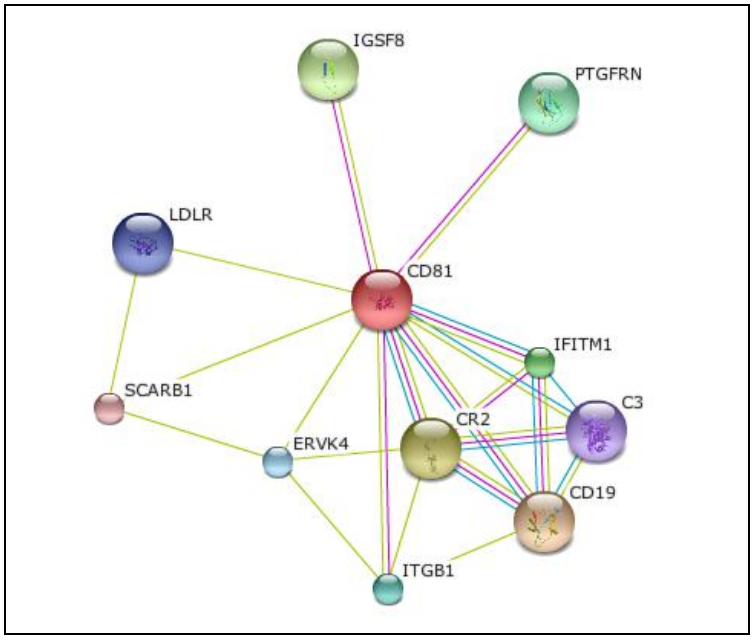

Figura 7 – Apresentação da rede de interação da proteína (STRING, 2000)

2.4.3 BLAST

BLAST[3] (*Basic Local Alignment Search Tool*) é um conjunto de programas de busca de similaridade projetados para explorar todas as bases de dados de sequências disponíveis, independente se a consulta é uma proteína ou DNA. Os programas BLAST foram projetados para a velocidade, com um sacrifício mínimo de sensibilidade para o relacionamento de sequências distantes. As pontuações atribuídas em uma pesquisa BLAST têm uma interpretação estatística bem definida.

BLAST utiliza um algoritmo heurístico de buscas locais ao invés de alinhamentos globais e é capaz de detectar relacionamentos entre as seqüências que compartilham apenas regiões isoladas de similaridade. Começa uma busca pela indexação de todas as cadeias de caracteres de um determinado tamanho, em seguida verifica o banco de dados a procura de correspondências entre as "palavras" indexadas na busca e textos encontrados no banco de dados de sequências. Para pesquisas de nucleotídeo para nucleotídeo, essa busca deve ser exata, de proteína para proteína o valor das buscas é determinado usando uma matriz de substituição. Quando uma palavra for encontrada, duas palavras próximas no caso de pesquisas por proteínas, BLAST tenta estender a busca para frente e para trás a partir da

[3] BLAST. Disponível em: <http://blast.ncbi.nlm.nih.gov/Blast.cgi>. Acesso em 29 de março de 2011.

palavra para produzir um alinhamento. BLAST continuará procurando desde que a pontuação de alinhamento continue aumentando, ou até que fique com uma pontuação negativa dada por inadequação (BLAST Overview, 2007).

1) O processo de pesquisa do melhor alinhamento começa acessando o site do BLAST.

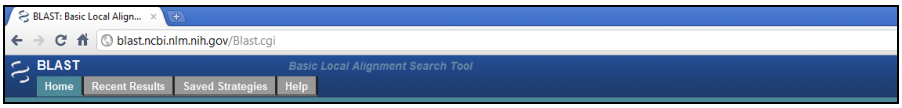

Figura 8 – Acesso ao site BLAST (BLAST, 2005)

2) Selecione o programa *protein blast* para efetuar a consulta conforme mostra a Figura 9.

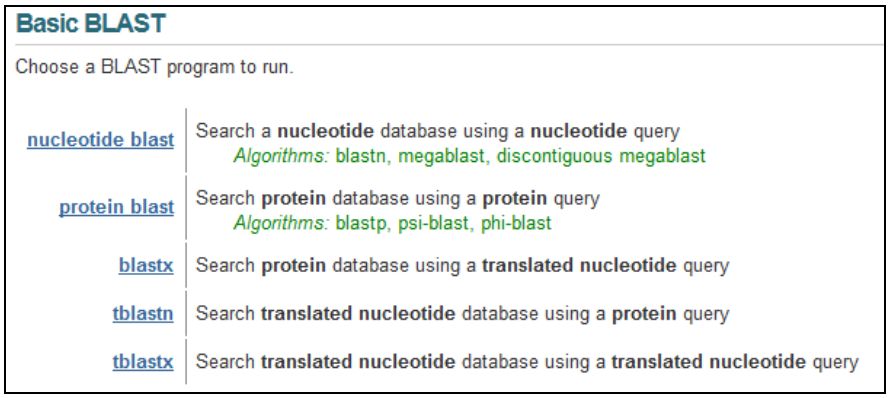

Figura 9 – Seleção do programa (BLAST, 2005)

3) No campo *Enter accession number(s), gi(s), or FASTA sequence(s)* podemos digitar o número do *GenInfo* ou seqüência da espécie.

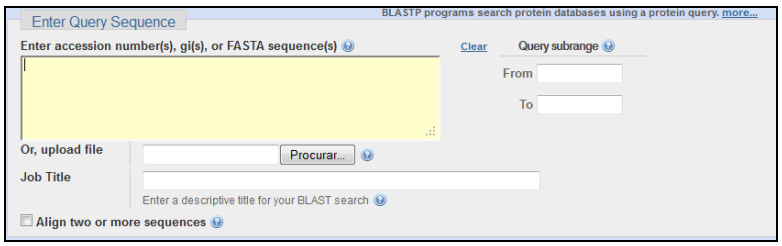

Figura 10 – Pesquisa da espécie (BLAST, 2005)

4) O site apresentará os resultados a seguir:

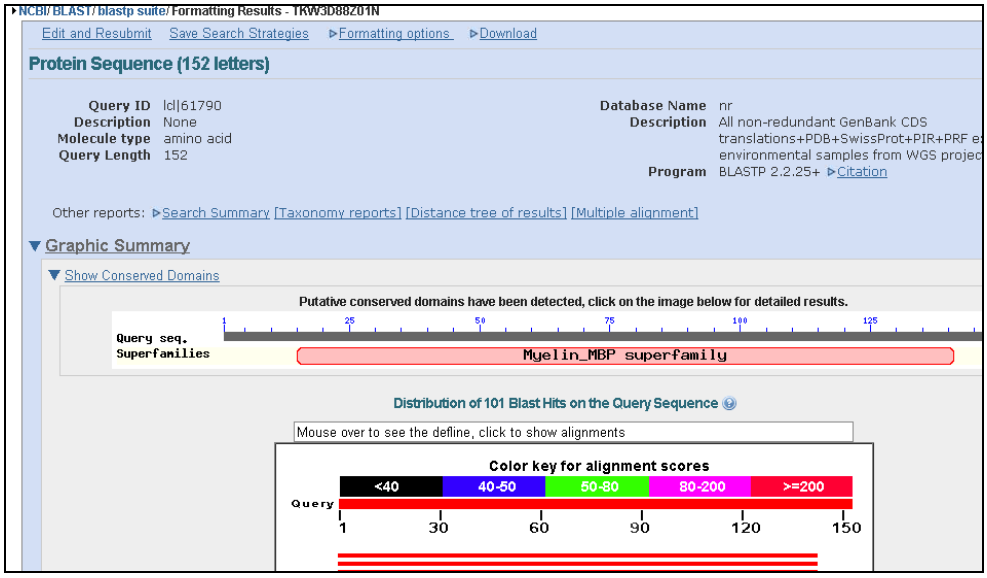

Figura 11 – Resultado da pesquisa da espécie (BLAST, 2005)

2.4.4 PathBLAST

PathBLAST[4] é a estratégia geral para o alinhamento de duas redes de interação protéica para elucidar suas vias conservadas. Este método identifica pares de caminhos de interação, elaborado a partir das redes de diferentes espécies ou de diferentes processos dentro de uma espécie, onde as proteínas em posições equivalentes compartilham forte sequência homológica.

A versão web pode ser consultada de duas maneiras, seja pela identificação (ID) ou pela seqüência. Identificadores (ID) válidos consistem de qualquer sinônimo válido encontrado no DIP (*Database of Interacting Proteins*). Se não for encontrado o identificador (ID) com a correspondência exata, serão apresentadas potenciais correspondências no banco de dados que poderão ser escolhidos. Se for escolhida a busca por sequência, então qualquer proteína válida é aceita. Se for informado o ID e a seqüência, o ID será ignorado e será usado apenas como um identificador. (NCBI, 2004)

1) O processo de pesquisa de uma proteína começa acessando o site do PathBLAST.

[4] PathBLAST – Disponível em: <http://www.pathblast.org>. Acesso em 29 de março de 2011.

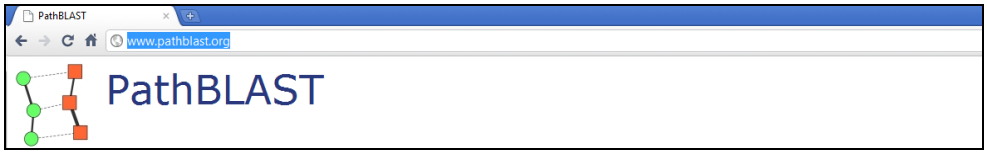

Figura 12 – Acesso ao site (PathBLAST, 2005)

2) No campo *Protein ID* digitamos a identificação da proteína e/ou no campo *Protein Sequence* digitamos a seqüência da proteína. No campo *Target Organism Network* selecionaremos o tipo de rede de organismos e clicamos no botão *BLAST*.

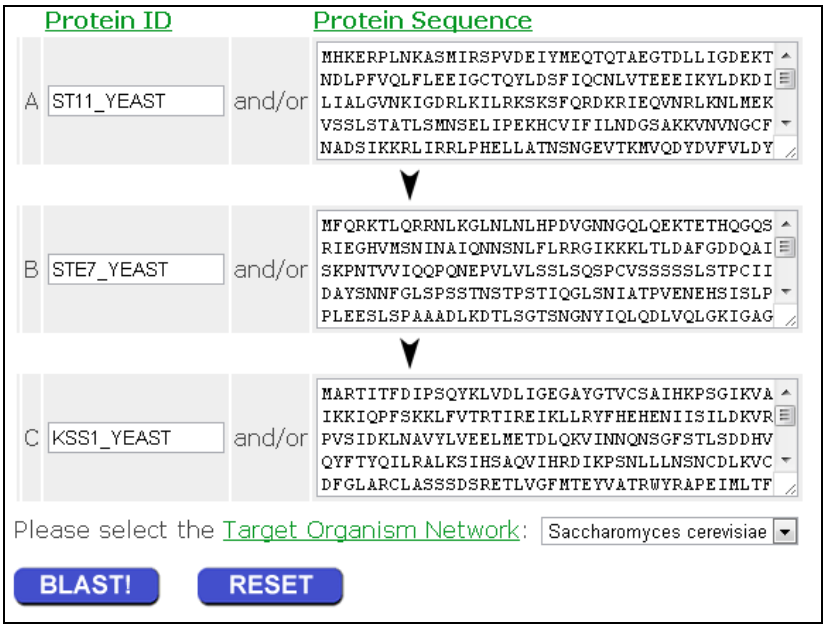

Figura 13 – Pesquisa de proteínas (PathBLAST, 2005)

3) O site lhe apresentará o identificador da requisição que será encaminhada ao site para processamento.

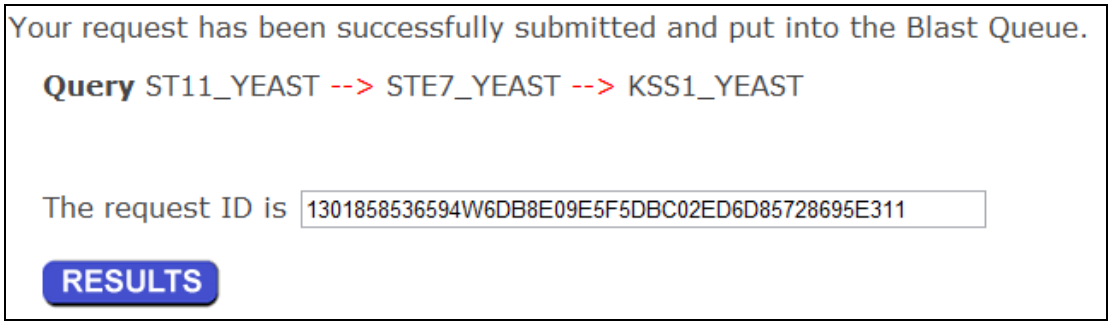

Figura 14 – Requisição para mostrar resultados (PathBLAST, 2005)

4) Ao clicarmos no botão *Results* o site retornará a espécie selecionada, o número de proteínas e o número de interações que foi encontrado nas redes com a devida pontuação.

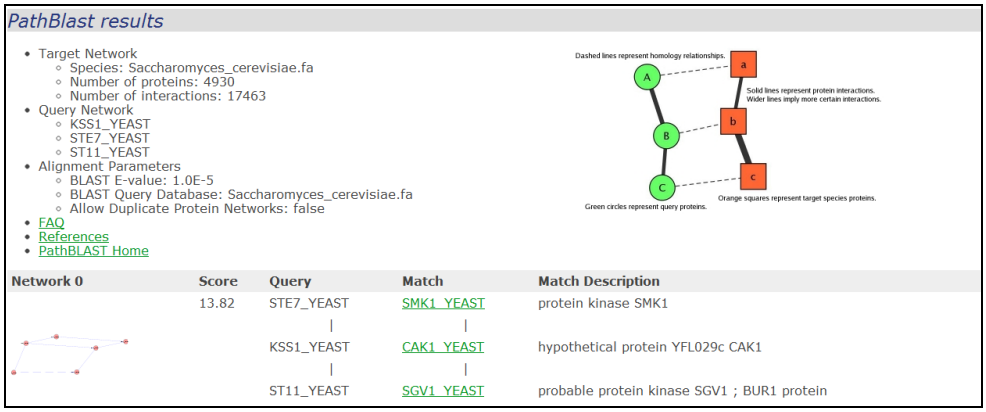

Figura 15 – Apresentação das redes encontradas (PathBLAST, 2005)

2.4.5 DIP

DIP[5] (*Database of Interacting Proteins*) é um banco de dados que fornece à comunidade científica uma ferramenta completa e integrada para a navegação e eficiente extração das informações sobre as interações protéicas e redes de interação em processos biológicos. O DIP foi criado para complementar os bancos de dados existentes e incluir proteínas que interagem a partir de muitos organismos permitindo aos cientistas expandir e complementar as observações de interações entre proteínas em um organismo com as observações de outros organismos.

2.5 Considerações finais

Nesse capítulo foram apresentados alguns conceitos da biologia molecular, alguns dos repositórios de dados biológicos existentes e que serão utilizados neste trabalho, o conteúdo disponibilizado pelos *sites* e exemplos de consulta de dados.

[5] DIP – Database of Interacting Proteins. Disponível em: <http://dip.doe-mbi.ucla.edu/dip/Main.cgi>. Acesso em 29 de março de 2011.

3 PROPOSTA DE SOFTWARE

O presente trabalho baseia-se no sistema desenvolvido por (Oldra, 2009). Utilizou-se a mesma lógica para as chamadas de urls, chamadas aos *web services* e foram necessários pequenos ajustes nas regras das expressões regulares. Recebeu melhorias de usabilidade, desempenho, possibilita o cadastro de proteínas a serem desconsideradas e principalmente o acesso ao site do PathBLAST. Este capítulo apresenta o sistema de consulta à base de dados biológicos desenvolvido anteriormente (Oldra, 2009) focando nos objetivos e em ideias para construção e desenvolvimento de consultas a redes de interação de proteínas a partir da extração de dados biológicos.

3.1 BioNet

O sistema *web* BioNet desenvolvido por Oldra (2009) é um sistema que facilita o processo de consulta dos usuários de bioinformática permitindo que sejam pesquisadas redes de interação de proteínas a partir de doenças gênicas, isso sem que seja necessário o uso direto das páginas *web* do OMIM e do STRING. O sistema se encarrega de fazer a comunicação com esses *sites* e ainda permite acompanhar todo o processo que está sendo realizado através de sua interface, podendo o usuário inclusive salvar e depois recuperar o processo executado (OLDRA, 2009).

O sistema foi desenvolvido com a linguagem PHP e necessita de uma máquina servidora. O servidor recebe as requisições de serviços e retorna a página que estará disponível em um diretório virtual do servidor *web*. Essa página permitirá ao usuário interagir com os outros serviços disponibilizados pelas páginas do OMIM e do STRING.

O fluxo de pesquisa do software para consulta à rede de interação de proteína será explicado a seguir usando o fluxograma apresentado na Figura 16.

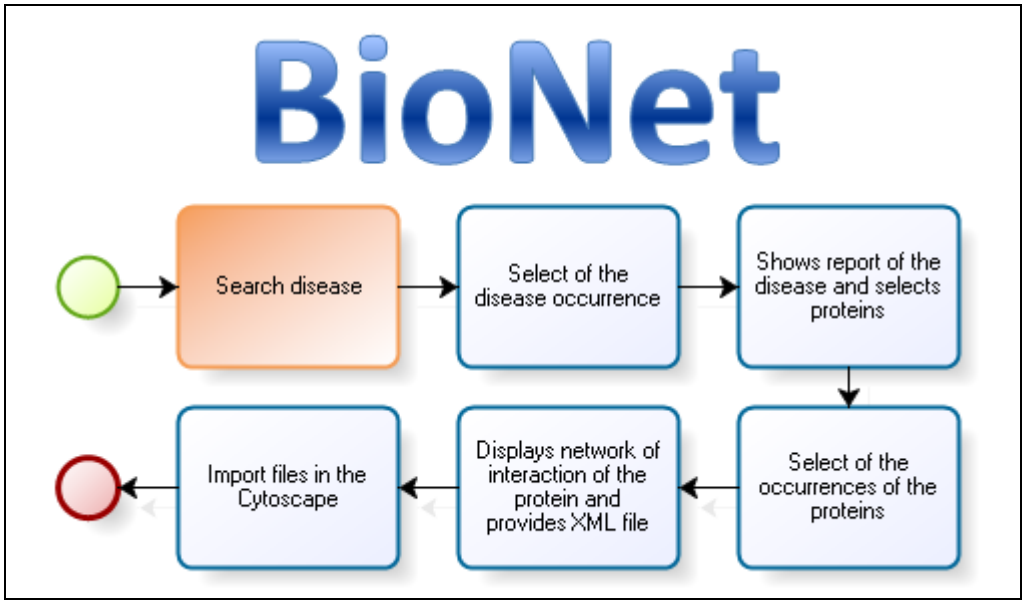

Figura 16 – Fluxo de pesquisa do software Bionet

O *workflow* científico da Figura 16 visa demonstrar as etapas que o especialista precisa realizar no sistema *web* para obter a(s) rede(s) de interação da(s) proteína(s). O processo no sistema se inicia com a pesquisa da doença, como pode ser visto na Figura 17. O usuário digita a doença que deseja encontrar e clica no botão *Search*.

Então o sistema apresenta as ocorrências de doenças com aquele termo, assim como poderia ser feito no *site* do OMIM, para que o usuário escolha a doença que deseja visualizar e clica sobre seu *link*, como mostra a Figura 18.

Após isso, o sistema irá apresentar o relatório da doença escolhida anteriormente (o mesmo apresentado pelo OMIM) e também irá sugerir algumas proteínas encontradas no relatório para que o usuário selecione as que deseja pesquisar, como mostra a Figura 19.

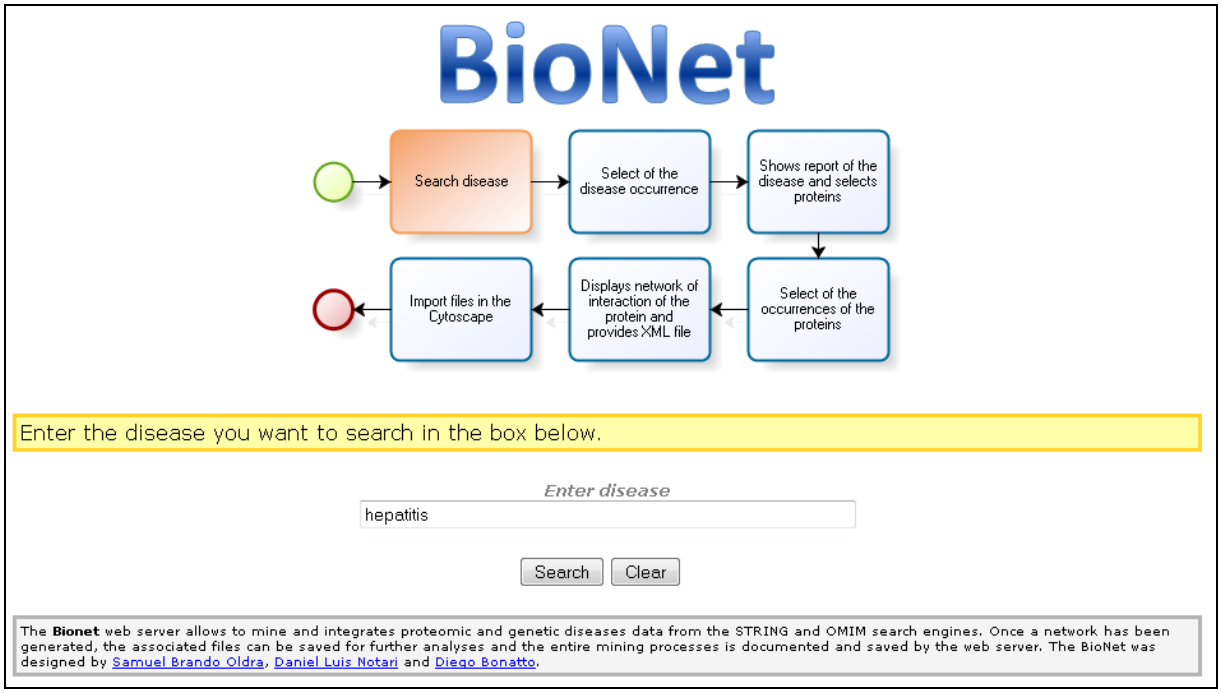

Figura 17 – Procura da doença

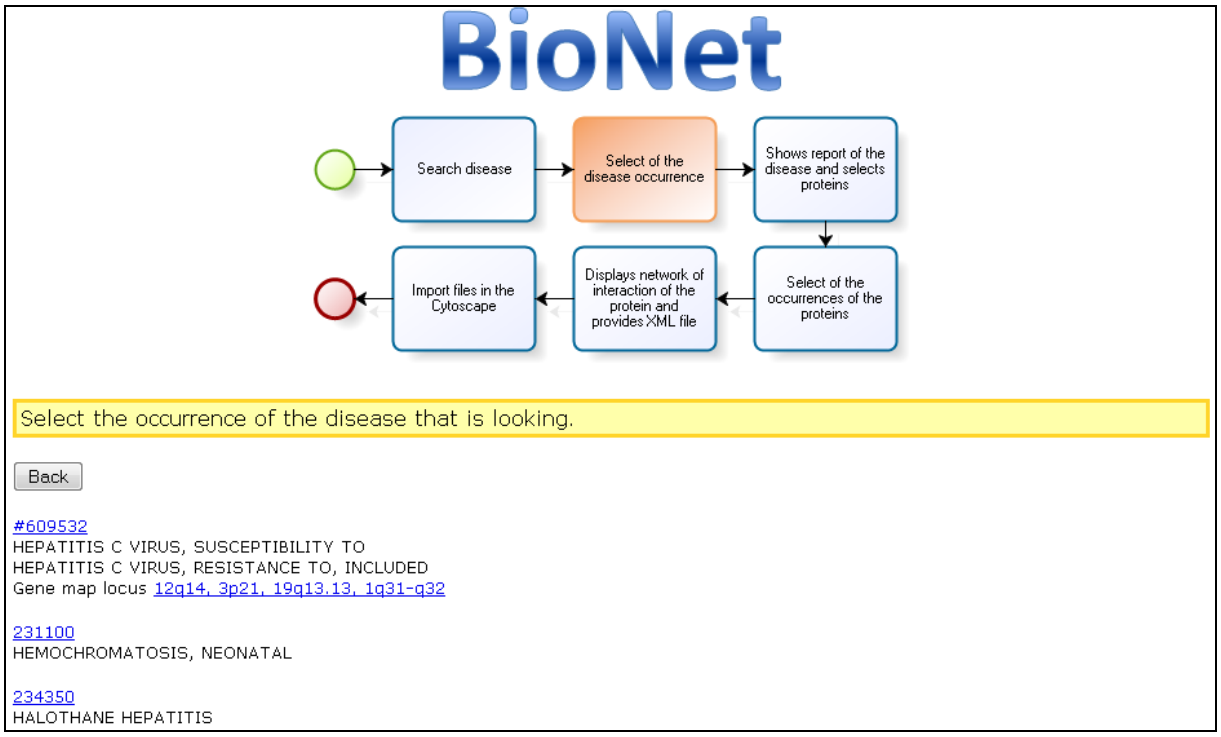

Figura 18 – Seleciona a ocorrência da doença

Então o usuário pode apagar as proteínas que não deseja pesquisar da *caixa de texto* e, acrescentar as que deseja pesquisar ou porque o sistema não encontrou ou porque não estão

no relatório da doença. Após isso o usuário clica no botão *Next* e o sistema irá pesquisar as ocorrências da(s) proteína(s).

Na sequência, o sistema irá apresentar as ocorrências de cada proteína selecionada (em humanos), como mostra a Figura 20, de forma como são apresentadas no *site* do STRING. O usuário então seleciona as que deseja visualizar na rede de interação da(s) proteína(s) e clica no botão *Next* novamente.

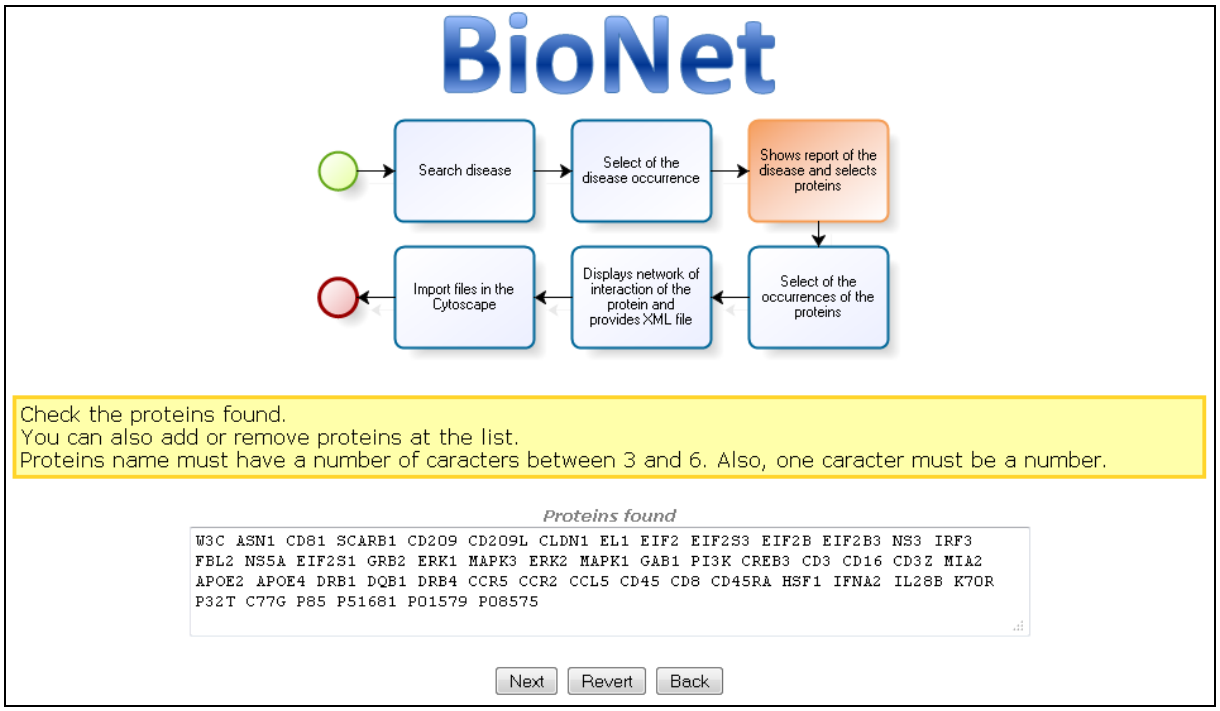

Figura 19 – Apresenta as proteínas encontradas

Por fim, o sistema apresenta a imagem da rede de interação da(s) proteína(s), como pode ser visto na Figura 21, e deixa disponível para *download* o arquivo XML clicando no *link Download XML*, que pode ser usado no software Cytoscape[6]. O sistema possibilita também a visualização dos outros arquivos que podem ser usados clicando no *link Other files*. Clicando sobre a imagem, o usuário será direcionado para a página do STRING para poder realizar qualquer alteração necessária.

[6] Cytoscape. Software de bioinformática para visualização de redes de interação molecular. Disponível em: <http://www.cytoscape.org>

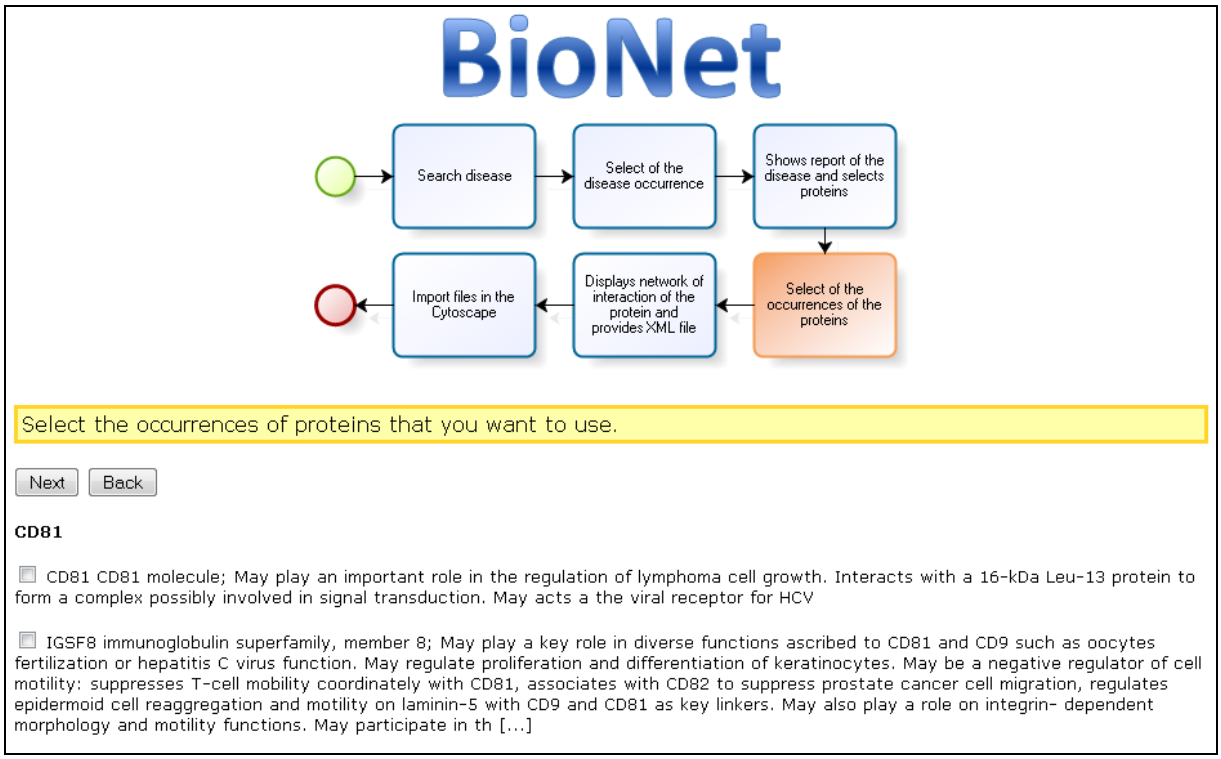

Figura 20 – Seleciona as ocorrências das proteínas

Então o usuário pode fazer outras pesquisas na seqüência, limpar o fluxo clicando no *link Clean flow*, salvar o processo clicando com o botão direito do *mouse* no *link Download flow* e escolher um local para o arquivo, adicionar ao fim do fluxo um outro executado anteriormente e/ou recuperar um fluxo clicando no botão *Choose...*, localizar o arquivo XML do fluxo e após clicando no botão *Load*, adicionar comentários ao fluxo digitando a mensagem na caixa de texto *User comments*, e após clicando no botão *Add*, e apagar etapas ou comentários desnecessários no fluxo clicando sobre o *link remove* ao lado do mesmo.

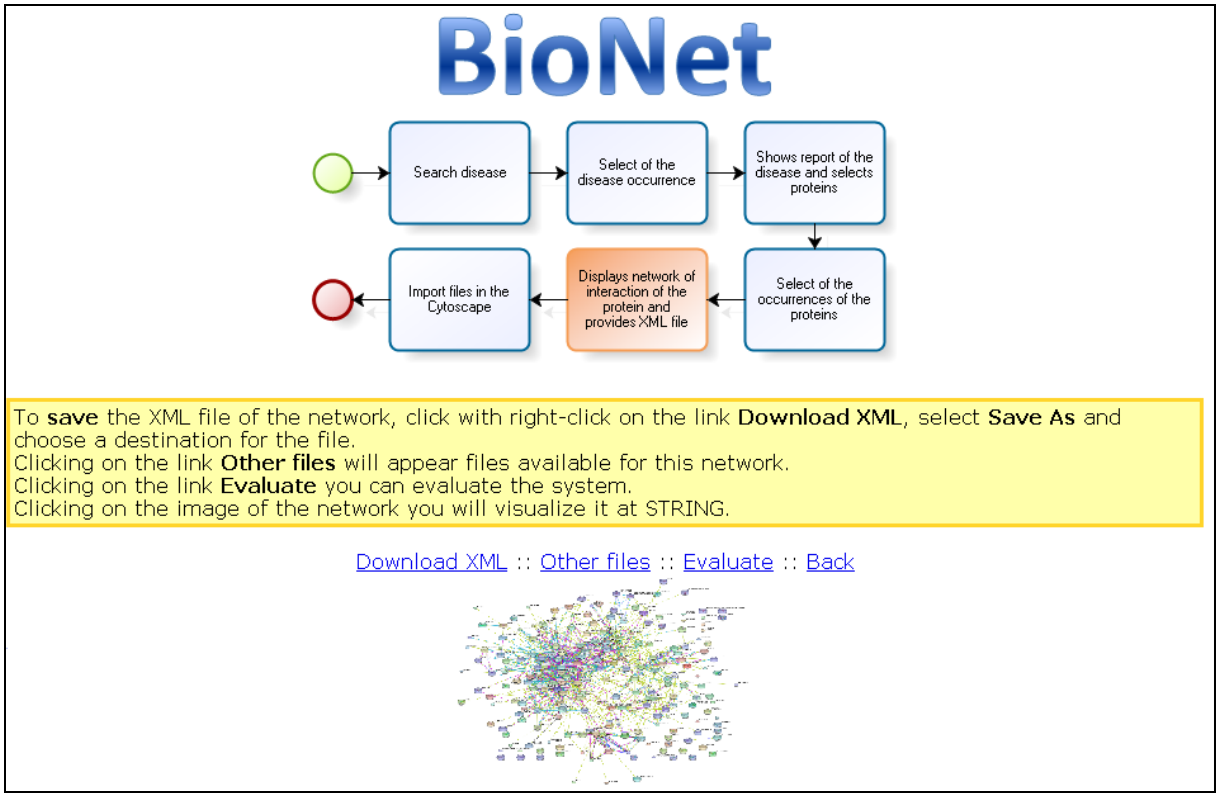

Figura 21 – Apresenta rede de interação da proteína e fornece arquivo XML

Abaixo serão listados os pontos positivos do sistema *web* do BioNet e os pontos que poderiam ser melhorados no sistema que será implementado neste trabalho:

PONTOS POSITIVOS

- Centralização das pesquisas dos sites OMIM e STRING em um único sistema.
- Possibilidade de salvar e carregar a pesquisa do usuário através de um arquivo XML.
- Fácil visualização do fluxo de pesquisa.

PONTOS A MELHORAR

- A consulta ao site do STRING é sempre realizada com a espécie *Homo Sapiens* (9606).

- A lista de proteínas que serão removidas (desconsideradas) está fixa no código fonte.
- As proteínas que não são encontradas no site do STRING são apresentadas como alertas (warning).

3.2 Novo workflow científico para consulta a redes de interação de proteínas

O sistema tem o objetivo de facilitar a consulta a redes de interação protéica possibilitando visualizar informações da pesquisa durante o processo para posterior utilização em novas pesquisas. Nessa seção serão apresentados os diagramas envolvidos com o objetivo de modelar o sistema desktop e melhorar o entendimento quanto ao funcionamento do mesmo.

O fluxo de pesquisa do software para consulta à rede de interação de proteína será explicado a seguir usando como base o fluxograma apresentado, conforme mostra a Figura 22. No primeiro passo do processo o usuário entrará com o nome da doença e solicitará a busca, no segundo passo será apresentada a lista de doenças encontradas com o nome da doença especificado.

No terceiro passo, ao selecionar a doença desejada o sistema efetuará uma nova busca das proteínas relacionadas a essa doença. No quarto passo o usuário poderá selecionar o organismo desejado e selecionar as ocorrências encontradas para as proteínas selecionadas anteriormente. No quinto passo o usuário poderá visualizar a imagem da rede, o arquivo texto e o arquivo XML da rede. No sexto passo o usuário selecionará até três proteínas para consultar o alinhamento da rede e visualizará o resultado no sétimo passo.

No oitavo passo o usuário poderá importar o arquivo XML apresentado no quinto passo, visualizar a rede no nono passo, no décimo gerar e analisar os agrupamentos e no décimo primeiro efetuar a partição das redes de interações.

O fluxo de pesquisa sempre seguirá as etapas listadas abaixo, pois depende do resultado da etapa anterior para prosseguir as consultas seguintes.

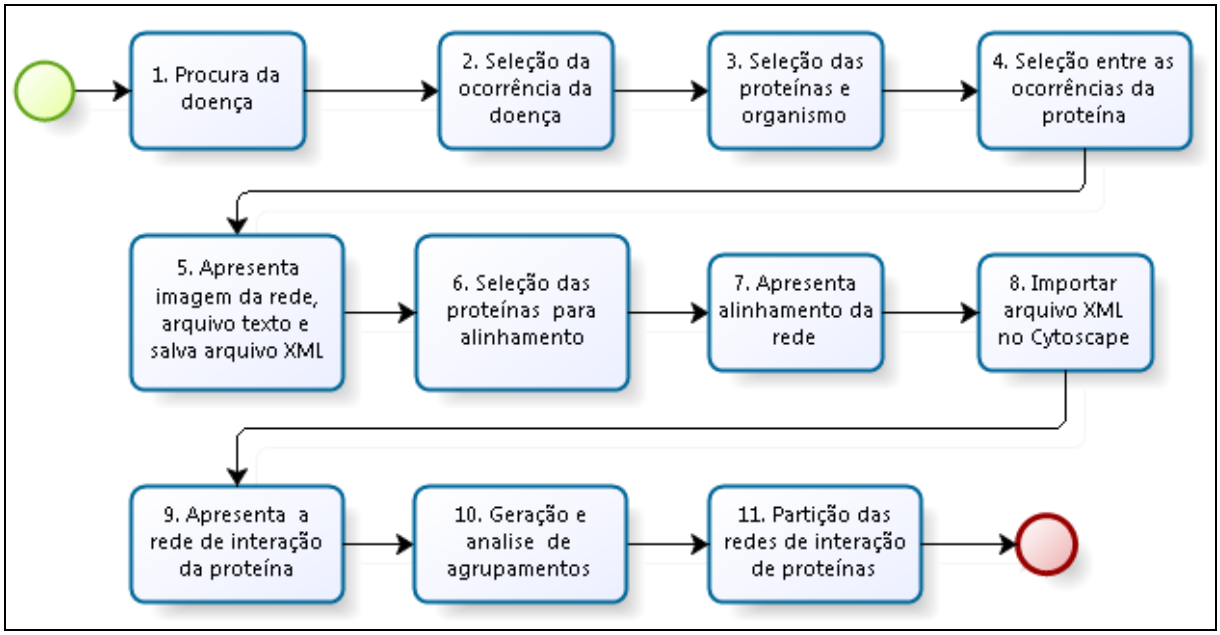

Figura 22 – Fluxo de pesquisa do software Bionet 2.0

3.3 Arquitetura da aplicação

Como pode ser observado na Figura 23 o sistema *desktop* permitirá ao usuário interagir com os serviços disponibilizados pelas páginas do OMIM, STRING e PathBLAST. A aplicação foi desenvolvida na linguagem de programação Java e necessita a instalação do software Java Runtime Environment (JRE) para que possa ser executado.

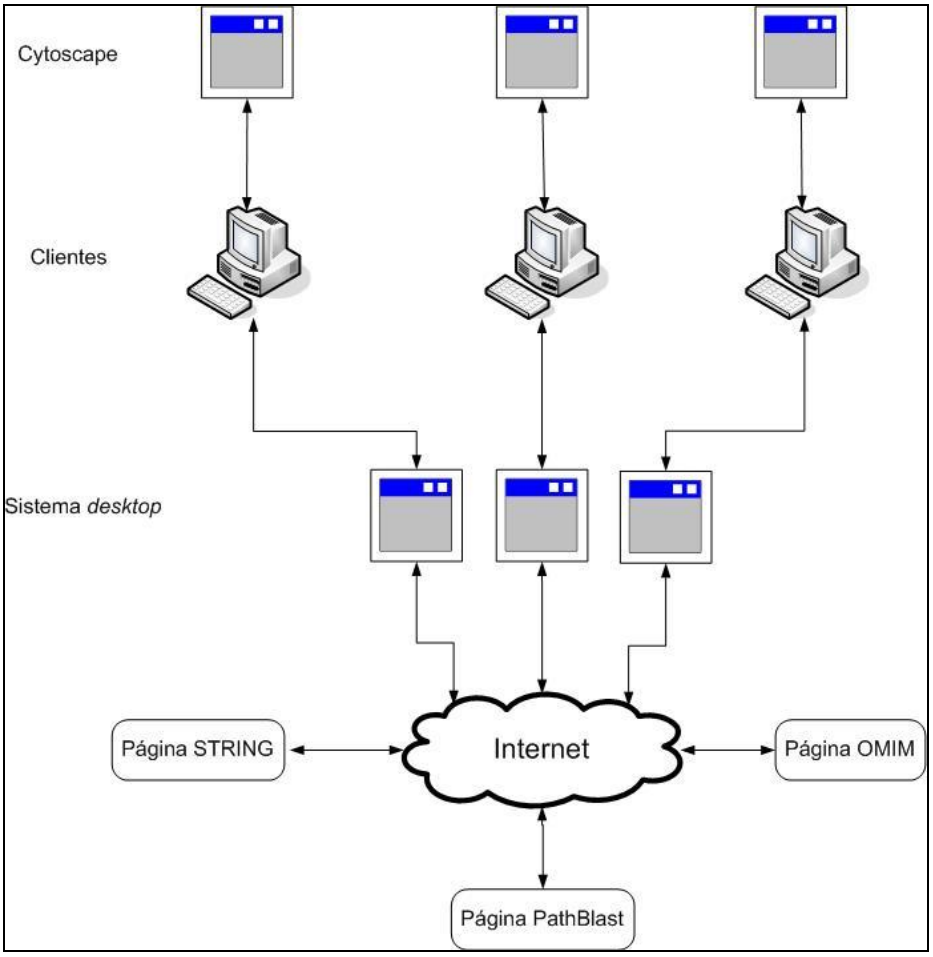

Figura 23 – Arquitetura da aplicação

3.4 Requisitos do sistema

No processo de pesquisa da doença, o usuário entrará com o nome da doença e o sistema efetuará a pesquisa no *site* do OMIM retornando a lista de doenças encontradas com o termo especificado.

No processo de pesquisa da proteína serão apresentadas as proteínas que possuem relação com a doença pesquisada anteriormente novamente através do *site* OMIM que retornará a lista de proteínas encontradas.

No processo de pesquisa da rede serão apresentadas as ocorrências das proteínas selecionadas e através do *site* STRING serão retornadas as informações das ocorrências.

No processo de visualização da rede serão apresentados os arquivos XML, a imagem da rede e o arquivo texto com informações das ocorrências através do *site* STRING.

No processo de visualização do alinhamento da rede serão selecionadas até três proteínas e a consulta será submetida ao *site* do PathBLAST que apresentará o alinhamento encontrado para as proteínas. O diagrama de caso de uso pode ser visualizado conforme mostra a Figura 24.

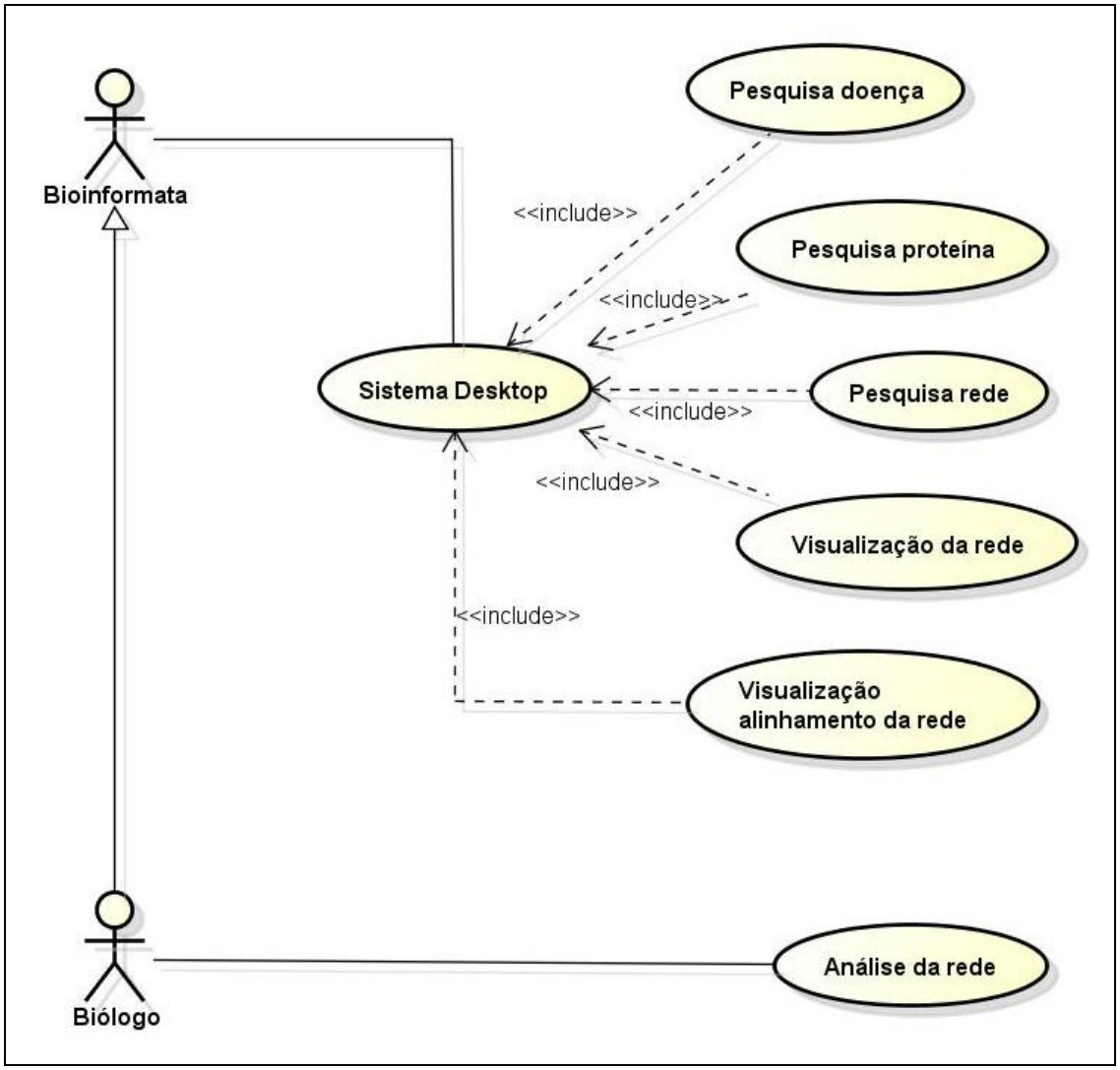

Figura 24 – Diagrama de caso de uso

3.5 Arquitetura interna do sistema

O diagrama de pacotes ilustra a arquitetura de um sistema mostrando o agrupamento de suas classes, descrevendo os pacotes ou pedaços do sistema divididos em agrupamentos lógicos que mostram as dependências entre estes, ou seja, pacotes podem depender de outros pacotes. O pacote é o elemento básico organizador de um modelo de sistema UML. Uma vez que representa um agrupamento, um pacote é em geral dono de diversos elementos: classes, interfaces, componentes, colaborações, casos de uso, diagramas, e até outros pacotes.

A representação do diagrama de pacotes da aplicação desenvolvida é apresentada a seguir, conforme mostra a Figura 25.

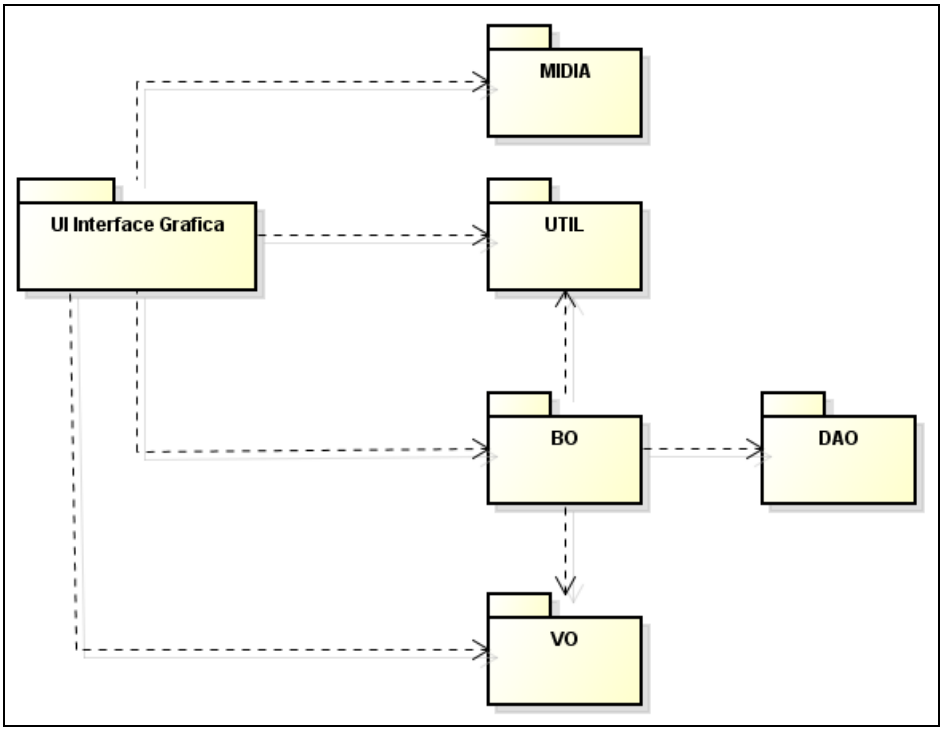

Figura 25 – Diagrama de pacotes

O sistema *desktop* é dividido em seis (6) pacotes, onde cada pacote é responsável por uma funcionalidade ou um grupo de funções:

- BO: Representam as classes que contêm a camada de negócio da aplicação. Também chamados de Business Object.

- DAO: Representam as classes que contêm a camada de acesso aos dados de banco de dados relacionais ou objetos. Também chamados de Data Access Object.

- MIDIA: Pacote que contêm as imagens utilizadas no sistema.

- UI: Representam as classes que contêm as interfaces disponibilizadas ao usuário para entrada e saída de dados.

- UTIL: Representam as classes auxiliares para processamento de texto, chamada de programas externos, processamento de expressões regulares, criação do log.

- VO: Representam as classes que realizam o transporte de objetos entre as camadas da aplicação.

3.6 Diagrama de classes

O diagrama de classes representa a estrutura do sistema, recorrendo ao conceito de classe e suas relações. O modelo de classes resulta de um processo de abstração onde são identificados os objetos relevantes do sistema. Cada classe é descrita através do seu nome, identificação de todos os atributos e identificação de todas as operações que traduzem seu comportamento.

Os diagramas de classes são chamados de diagramas “estáticos” porque mostram as classes, com seus métodos e atributos bem como os relacionamentos estáticos entre elas: quais classes “conhecem” quais classes ou quais classes “são parte” de outras classes, mas não mostram a troca de mensagens entre elas.

As classes que representam os principais pacotes criados no sistema *desktop* serão apresentadas a seguir, conforme mostra a Figura 26. As classes da camada de negócio podem ser visualizadas conforme mostra a Figura 27. As classes de acesso aos dados podem ser visualizadas conforme mostra a Figura 28. As classes auxiliares do sistema podem ser visualizadas na Figura 29. As classes que realizam o transporte de objetos e a interface gráfica para interação com o usuário podem ser visualizadas conforme mostra a Figura 30 e Figura 31, respectivamente.

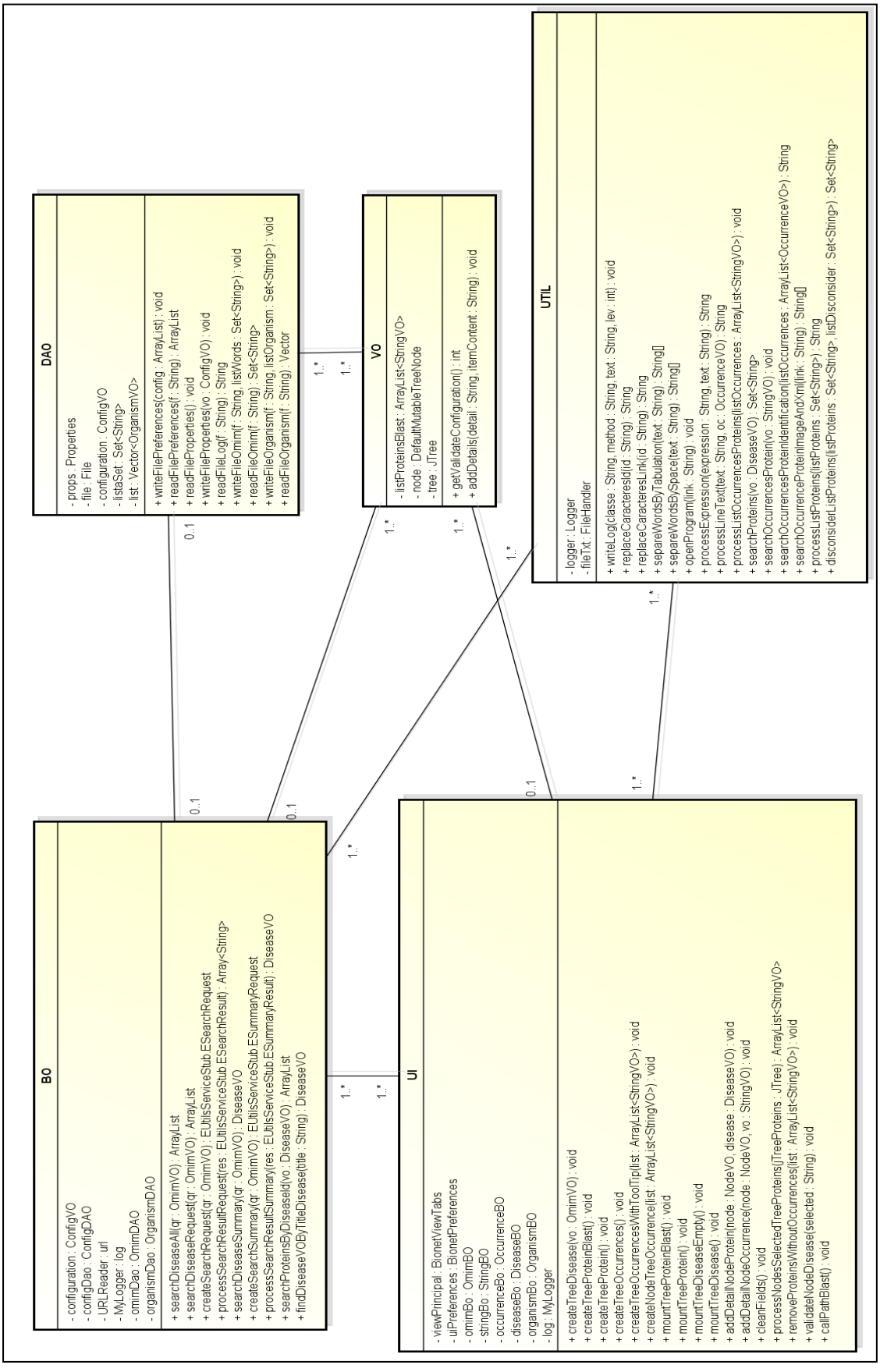

Figura 26 – Diagrama de classes – Bionet 2.0

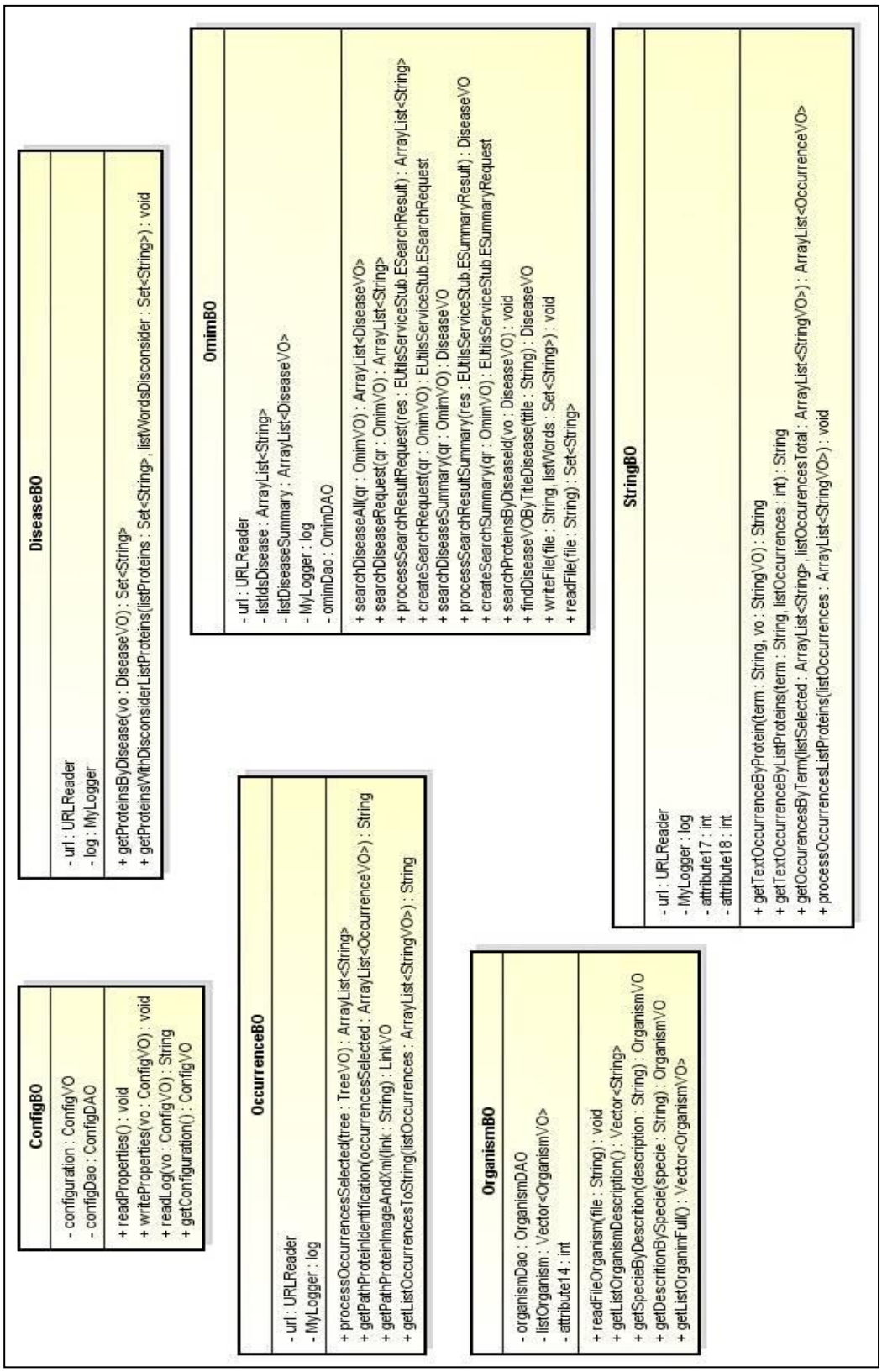

Figura 27 – Diagrama de classes – BO

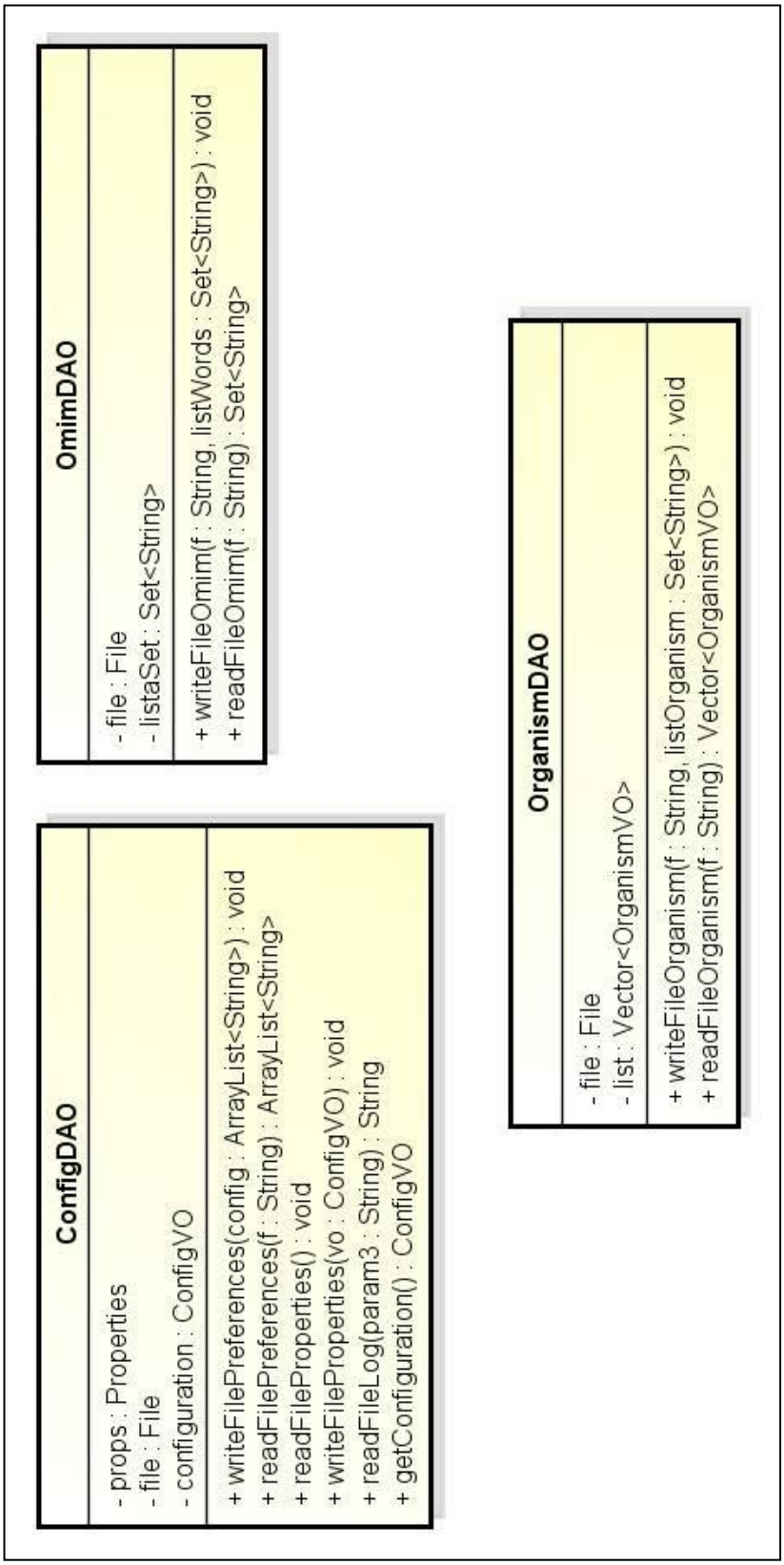

Figura 28 – Diagrama de classes – DAO

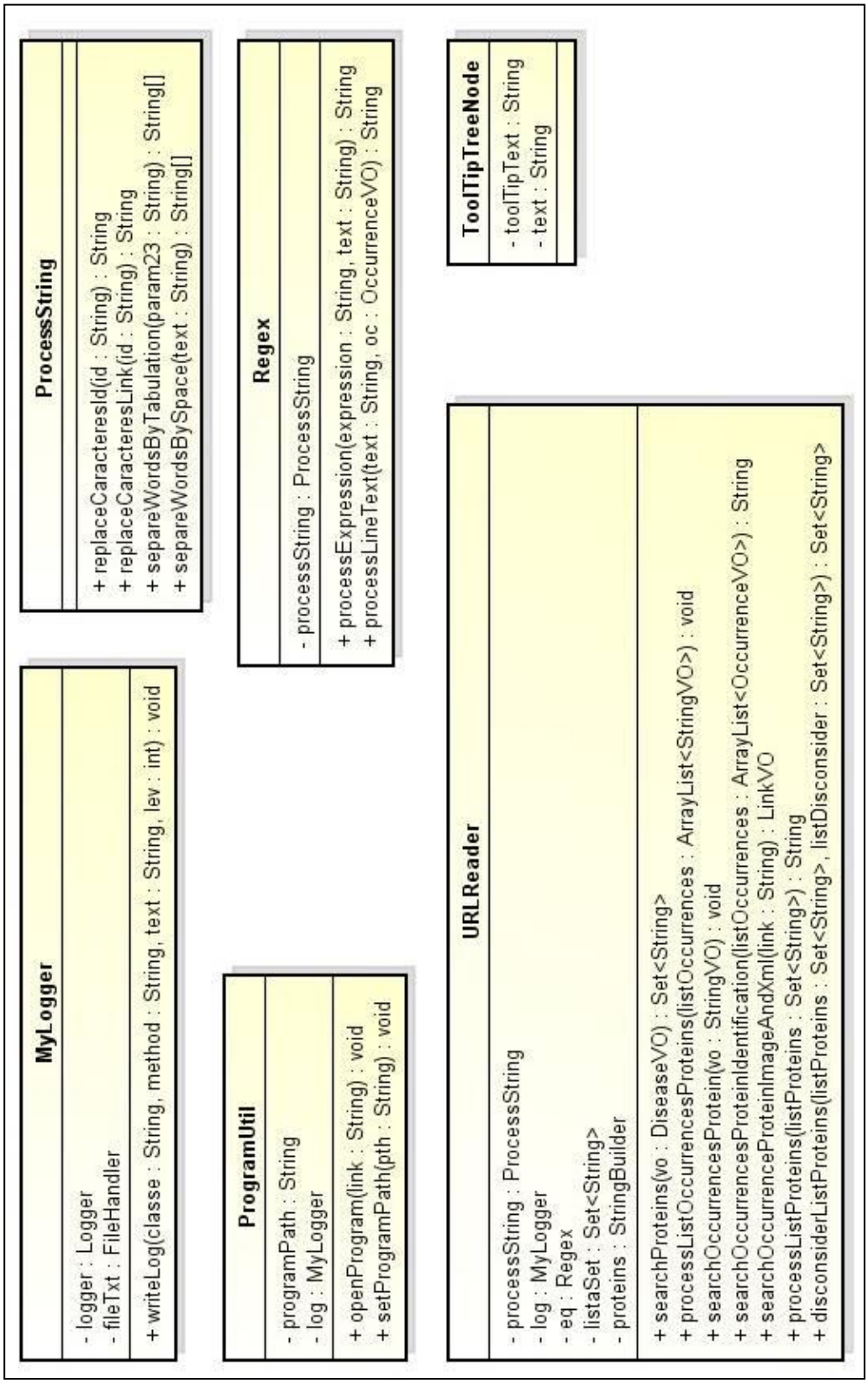

Figura 29 – Diagrama de classes – UTIL

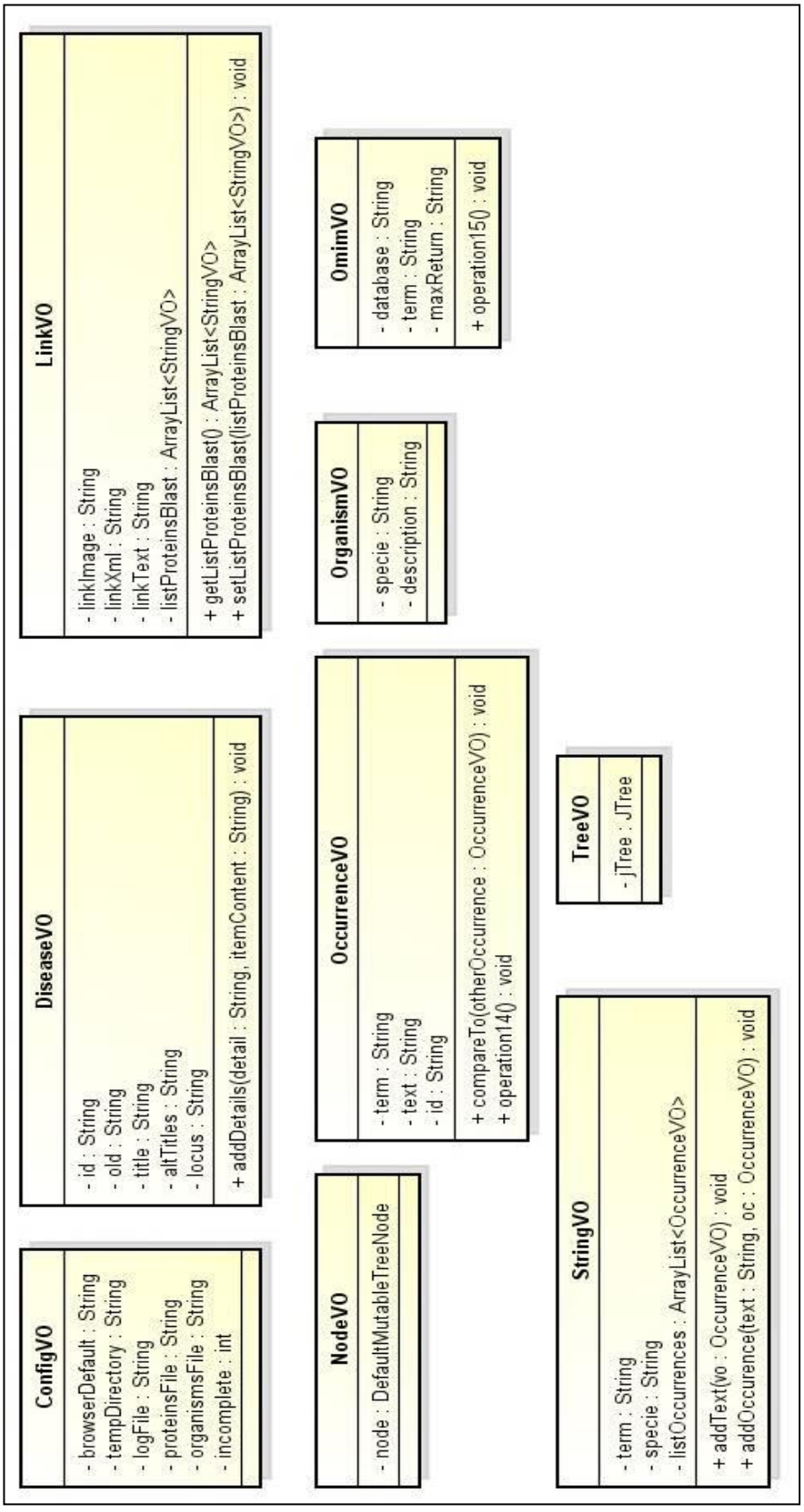

Figura 30 – Diagrama de classes – VO

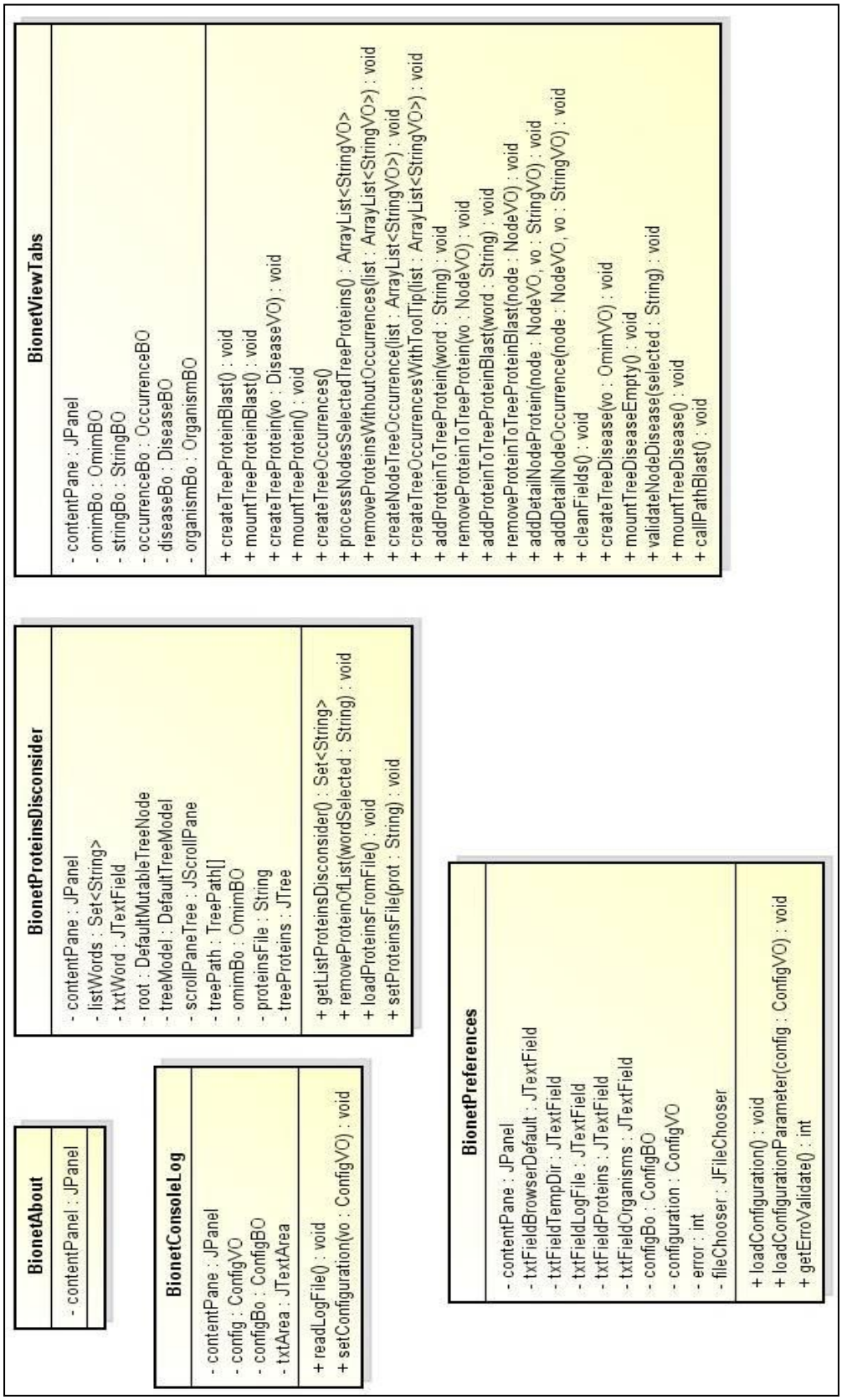

Figura 31 – Diagrama de classes – UI

3.7 Considerações finais

Nesse capítulo foi apresentado o software desenvolvido previamente (Oldra, 2009) com o seu devido fluxo proposto. Foi apresentado também a nova arquitetura da aplicação, o workflow que foi desenvolvido no presente trabalho com melhorias de usabilidade e a nova arquitetura do sistema para consulta aos site de dados biológicos.

4 IMPLEMENTAÇÃO

Nesse capítulo serão explicados as classes Java implementadas e o novo workflow científico do sistema.

O sistema de consulta a redes de interação de proteínas facilita o processo dos usuários de bioinformática permitindo que sejam pesquisadas redes de interação de proteínas a partir de doenças gênicas, isso sem que seja necessário o uso direto das páginas web do OMIM, STRING e PathBLAST. A aplicação se encarrega de fazer a comunicação com esses sites e permite acompanhar todo o processo que está sendo realizado através de sua interface.

Nas subseções que seguem será apresentado o que faz cada uma das principais classes, apresentado os principais trechos de código responsáveis pelas funcionalidades do sistema. A implementação refletirá os passos do *workflow* apresentando anteriormente.

4.1 Pesquisa da doença

Na primeira aba (*Step* 1) da aplicação o usuário poderá digitar o nome da doença que deseja encontrar e solicitará a pesquisa clicando no botão *Search*, conforme mostra a Figura 32.

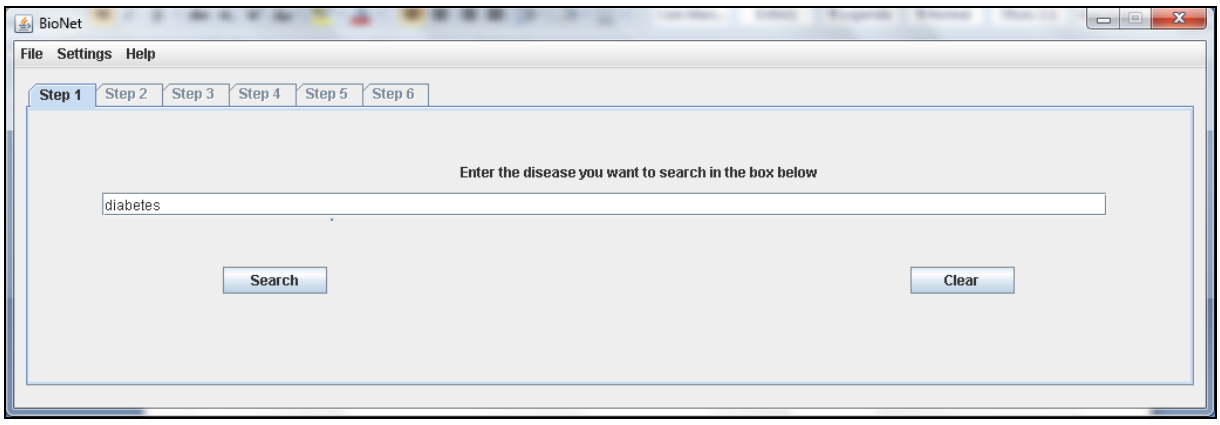

Figura 32 – Sistema *desktop* – *Step* 1

O evento vinculado ao botão *Search* validará se o usuário digitou alguma informação no campo texto (linha 297) e criará um objeto do tipo *OmimVO* com o nome da doença

digitada pelo usuário (linha 298). Em seguida, executará o método *createTreeDisease* que receberá como parâmetro o objeto *OmimVO* (linha 300), conforme mostra a Figura 33.

```
295     btnSearch.addActionListener(new ActionListener() {
296        public void actionPerformed(ActionEvent arg0) {
297           if (txtFieldDisease.getText().length() > 0) {
298              OmimVO vo = new OmimVO("omim", txtFieldDisease.getText(), null);
299              startProcessMouse();
300              createTreeDisease(vo);
301              stopProcessMouse();
302              enableTabbedPane(1, true);
303           }
304        }
305     });
```

Figura 33 – Método vinculado ao botão *Search*

Ao iniciar o método *createTreeDisease* um nodo pai será criado para agrupar a lista de doenças (linha 953) que será pesquisada e por fim executará o método *searchDiseaseAll* da classe *OmimBO* (linha 954). O método *searchDiseaseAll* retornará um *ArrayList* de objetos do tipo *DiseaseVO*. Para cada um dos objetos percorridos um novo nodo filho será criado (linha 959), setado com a descrição da doença e adicionado ao nodo pai (linha 962), conforme mostra a Figura 34.

```
952   private void createTreeDisease(OmimVO vo) {
953      rootDisease = new DefaultMutableTreeNode(vo.getDatabase());
954      listDisease = omimBo.searchDiseaseAll(vo);
955      if (listDisease != null) {
956         for (int i = 0; i < listDisease.size(); i++) {
957           DiseaseVO disease = new DiseaseVO();
958           disease = (DiseaseVO) listDisease.get(i);
959           DefaultMutableTreeNode nodeTree = new DefaultMutableTreeNode(disease.getTitle());
960           NodeVO node = new NodeVO(nodeTree);
961           this.addDetailNodeProtein(node, disease);
962           rootDisease.add(nodeTree);
963         }
964         this.mountTreeDisease();
965      } else {
966               this.mountTreeDiseaseEmpty();
967      }
968   }
```

Figura 34 – Método para montagem da árvore de doenças

Ao iniciar o método *searchDiseaseAll* o programa executará o método *searchDiseaseRequest* (linha 33) que efetuará a consulta ao banco OMIM através do *Web Service* disponível no site e em seguida executará o método *searchDiseaseSummary* que buscará detalhes da doença encontrada (linha 40), conforme mostra a Figura 35.

```
31  public ArrayList<DiseaseVO> searchDiseaseAll(OmimVO qr){
32      log.writeLog(this.getClass().getName(), this.getNameMethod(), "Database: " +
qr.getDatabase() + " Term: " + qr.getTerm() + " Max. Return: " + qr.getMaxReturn(), 2);
33      listIdsDisease = searchDiseaseRequest(qr);
34      if (listIdsDisease != null){
35          if (listIdsDisease.size() > 0){
36              listDiseaseSummary = new ArrayList<DiseaseVO>();
37              for (int i = 0; i < listIdsDisease.size(); i++ ){
38                  String id = listIdsDisease.get(i).toString();
39                  OmimVO queryDisease = new OmimVO(qr.getDatabase(), id, null);
40                  DiseaseVO disease = searchDiseaseSummary(queryDisease);
41                  listDiseaseSummary.add(disease);
42              }
43          } else{ listDiseaseSummary = null;}
44      }
45      return listDiseaseSummary;
46  }
```

Figura 35 – Método de busca dos detalhes da doença

O método *searchDiseaseRequest* buscará a doença digitada através da ação *run_eSearch* (linha 52) que retornará o código de identificação (ID) das doenças. Todos os código encontrados serão armazenados em um *ArrayList* que será retornado pelo método *processSearchResultRequest* (linha 53), conforme mostra a Figura 36.

```
48  public ArrayList<String> searchDiseaseRequest(OmimVO qr){
49      try{
50          EUtilsServiceStub service = new EUtilsServiceStub();
51          EUtilsServiceStub.ESearchRequest req = createSearchRequest(qr);
52          EUtilsServiceStub.ESearchResult res = service.run_eSearch(req);
53          listIdsDisease = processSearchResultRequest(res);
54          return listIdsDisease;
55      }  catch (Exception e) {
56       log.writeLog(this.getClass().getName(), this.getNameMethod(), e.toString(), 0);
57          return null;
58      }
59  }
```

Figura 36 – Método *run_eSearch* do site OMIM

Com base no código de identificação de cada uma das doenças o sistema executará um nova consulta pelo método *searchDiseaseSummary* e pela ação *run_eSummary* (linha 83) ao *Web Service* buscando detalhes da doença, como por exemplo, descrição da doença, descrição alternativa e outras informações, conforme mostra a Figura 37.

```
79  public DiseaseVO searchDiseaseSummary(OmimVO qr){
80     try {
81        EUtilsServiceStub service = new EUtilsServiceStub();
82        EUtilsServiceStub.ESummaryRequest req = createSearchSummary(qr);
83        EUtilsServiceStub.ESummaryResult res = service.run_eSummary(req);
84        return processSearchResultSummary(res);
85     } catch(Exception e) {
86      log.writeLog(this.getClass().getName(), this.getNameMethod(), e.toString(), 0);
87        return null;
88     }
89  }
```

Figura 37 – Método *run_eSummary* do site OMIM

4.2 Busca e seleção da doença

Na segunda aba da aplicação (*Step* 2) o sistema apresentará o resultado da pesquisa da doença, possibilitando ao usuário escolher a doença que está procurando conforme Figura 38.

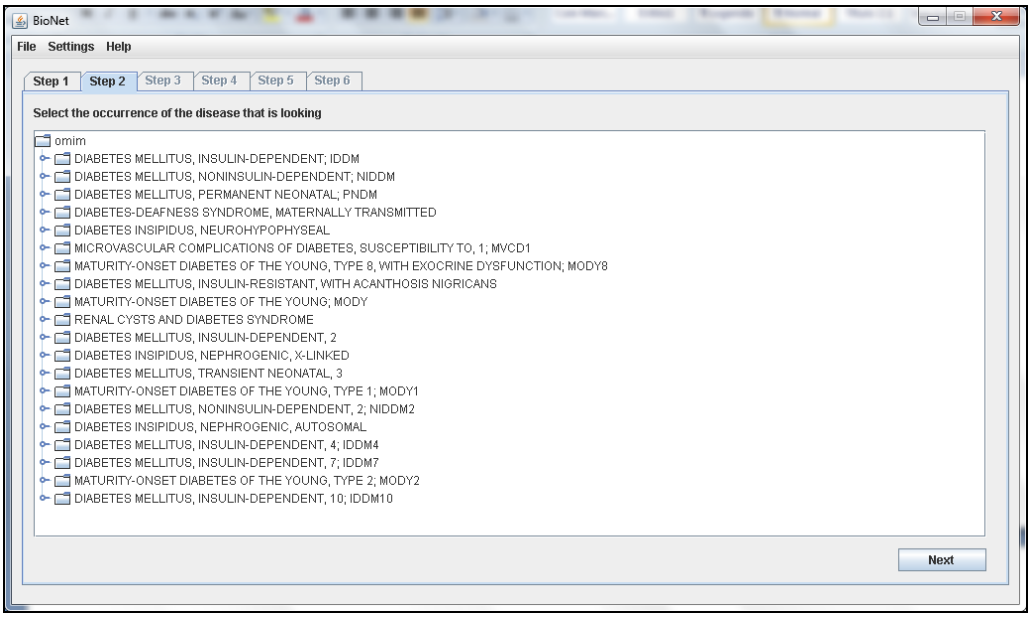

Figura 38 – Sistema *desktop* – *Step* 2

Ao selecionar a doença desejada e clicar no botão *Next*, o evento vinculado ao botão validará se a árvore de doenças foi criada no passo anterior (linha 332) e se algum nodo da árvore foi selecionado (linha 333). Com base no nodo selecionado (doença) será executado o método *validateNodeDisease* que validará se o item selecionado é o nodo pai ou se foi selecionado um nodo filho, após a validação será executado o método *findDiseaseVOByTitleDisease* que verificará no *ArrayList* qual doença apresenta o título selecionado e retornará um objeto *DiseaseVO* com os dados da doença (linha 339). Após isso, o sistema executará o método *createTreeProtein* (linha 341) que receberá o objeto *DiseaseVO* como parâmetro e será o responsável por montar a árvore de proteínas relacionadas à doença, conforme mostra a Figura 39.

```
330     btnStep2.addActionListener(new ActionListener() {
331         public void actionPerformed(ActionEvent arg0) {
332             if (jTreeDisease != null) {
333                 if (jTreeDisease.getLastSelectedPathComponent() != null) {
334                         String selected = jTreeDisease.getLastSelectedPathComponent().toString();
335                         boolean validate = validateNodeDisease(selected);
336                         if(!validate){
337                        JOptionPane.showMessageDialog(null, "The select node is a child node. Please
select the parent node!");
338                 } else{
339                     DiseaseVO vo = omimBo.findDiseaseVOByTitleDisease(selected);
340                     startProcessMouse();
341                     createTreeProtein(vo);
342                     enableTabbedPane(2, true);
343                     stopProcessMouse();
344                 }
345                 }
346             }
347         }
348     });
```

Figura 39 – Método de busca das proteínas da doença

Ao iniciar o método *createTreeProtein* um nodo pai será criado (linha 717) para agrupar a lista de proteínas encontradas para a doença selecionada. O sistema carregará também a lista de proteínas que devem ser desconsideradas (linha 718) e executará o método *getProteinsByDisease* que executará a consulta ao site do OMIM e processará o retorno da url de busca das proteínas do site (linha 719). Um exemplo da URL de busca das proteínas pode ser visualizado conforme mostra a Figura 40.

```
url = new URL("http://www.ncbi.nlm.nih.gov/omim/610551");
```

Figura 40 – Exemplo da URL de busca das proteínas

Ao retornar a lista *Set* de elementos não duplicados será executado o método *getProteinsWithDisconsiderListProteins* que eliminará da lista de proteínas *listProteins* as proteínas que foram previamente cadastradas na aplicação (linha 720). Para cada uma das proteínas que passarem pelo filtro anterior será criado um novo nodo (linha 725) e será adicionado à árvore de proteínas (linha 726) e por fim será executado o método *mountTreeProtein* que exibirá a árvore em tela (linha 729), conforme mostra a Figura 41.

```
716   private void createTreeProtein(DiseaseVO vo) {
717       rootProteins = new DefaultMutableTreeNode("Proteins");
718       listWordsDisconsider = uiProteinDisconsider.getListProteinsDisconsider();
719       listProteins = diseaseBo.getProteinsByDisease(vo);
720       diseaseBo.getProteinsWithDisconsiderListProteins(listProteins, listWordsDisconsider);
721       if (listProteins != null) {
722           for (Iterator<String> iter = listProteins.iterator(); iter.hasNext();) {
723               String protein = (String) iter.next();
724               log.writeLog(this.getClass().getName(), getNameMethod(), "Reading protein: " +
protein, 2);
725               DefaultMutableTreeNode node = new DefaultMutableTreeNode(protein);
726               rootProteins.add(node);
727           }
728       }
729       this.mountTreeProtein();
730   }
```

Figura 41 – Método de busca das proteínas da doença

4.3 Visualização e seleção das proteínas

Na terceira aba da aplicação (Step 3) o sistema apresentará as proteínas que foram encontradas para a doença na consulta ao *site* do OMIM e possibilitará selecionar qual o organismo que será utilizado na próxima pesquisa ao *site* do STRING. O organismo padrão utilizado na pesquisa ao *site* do STRING é *Homo Sapiens* que possui o código da espécie igual a 9606.

Nesta aba será disponibilizado uma caixa de texto e dois botões de inclusão e exclusão de proteínas. Para adicionar uma nova proteína o usuário digitará a proteína desejada e clicará

no botão "+". Para remover uma proteína, basta selecionar a proteína desejada e clicar no botão "-", conforme mostra a Figura 42.

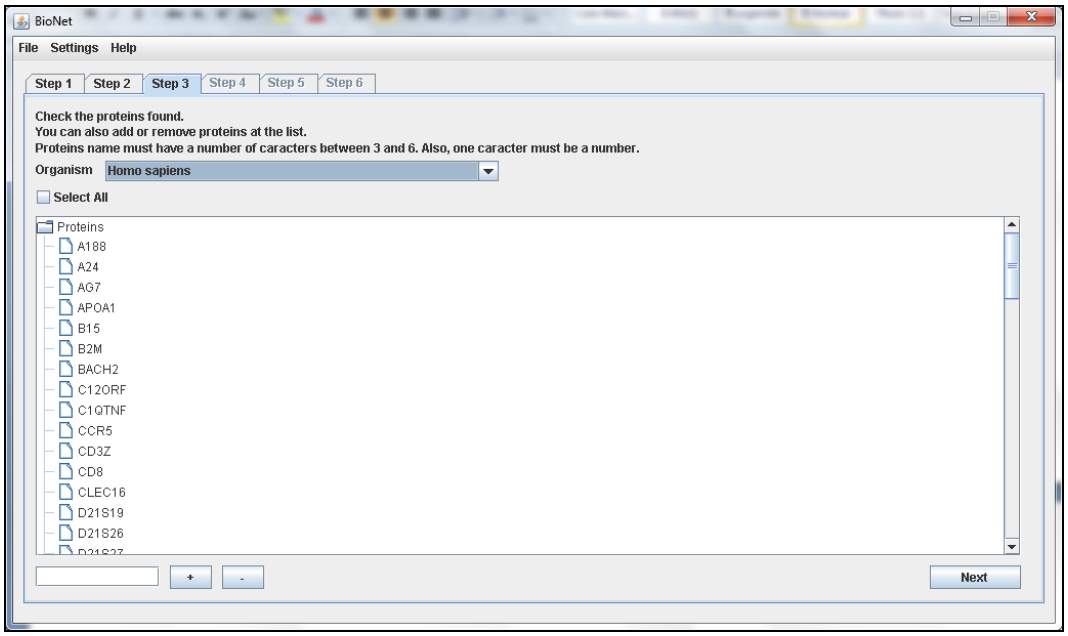

Figura 42 – Sistema *desktop* – *Step* 3

Ao clicar no botão *Next* será executado o método *createTreeOccurrences* que validará se existe uma lista de proteínas e se algum organismo foi selecionada (linha 745 e 746). O método *processNodesSelectedTreeProteins* será executado, a árvore de proteínas será percorrida e cada nodo selecionado será adicionado ao *ArrayList* (linha 747) e em seguida será processado pelo método *processOccurrencesListProteins* que executará a url de consulta ao site do STRING para cada uma das ocorrências selecionadas, conforme Figura 43.

```
739  private void createTreeOccurrences() {
740     treePathOccurrences = null;
741     listOccurrences = null;
742     rootOccurrences = new DefaultMutableTreeNode("Occurrences");
743     String descriptionOrganism = (String) jComboOrganism.getSelectedItem();
744     organism = organismBo.getSpecieByDescrition(descriptionOrganism);
745     if (listProteins != null && organism != null) {
746        if (listProteins.size() > 0 && organism.getSpecie() != null){
747           listOccurrences = processNodesSelectedTreeProteins(jTreeProteins);
748           stringBo.processOccurrencesListProteins(listOccurrences);
749           this.removeProteinsWithoutOccurrences(listOccurrences);
750           this.createTreeOccurrencesWithToolTip(listOccurrences);
751        }
752     }}
```

Figura 43 – Método de busca das ocorrências das proteínas

```
http://string-db.org/api/tsv-no-header/resolve?identifier=IL1&species=9606
```

Figura 44 – Exemplo da URL de consulta das ocorrências no site STRING

A url retornará um arquivo texto que será processado contendo as ocorrências relacionadas à proteína separando as informações e possibilitando vincular ao objeto *StringVO* as ocorrências lidas. Após a consulta das proteínas selecionadas todas as proteínas para as quais não forem encontradas ocorrências serão eliminadas do *ArrayList* de ocorrências pelo método *removeProteinsWithoutOccurrences* (linha 749) e as que possuírem ocorrências serão adicionadas ao nodo pai para que sejam apresentadas corretamente na árvore através do método *createTreeOccurrencesWithToolTip* (linha 750).

4.4 Seleção das ocorrências das proteínas

Na quarta aba da aplicação (Step 4) o sistema apresentará as proteínas com as devidas ocorrências encontradas no processamento anterior, conforme mostra a Figura 45.

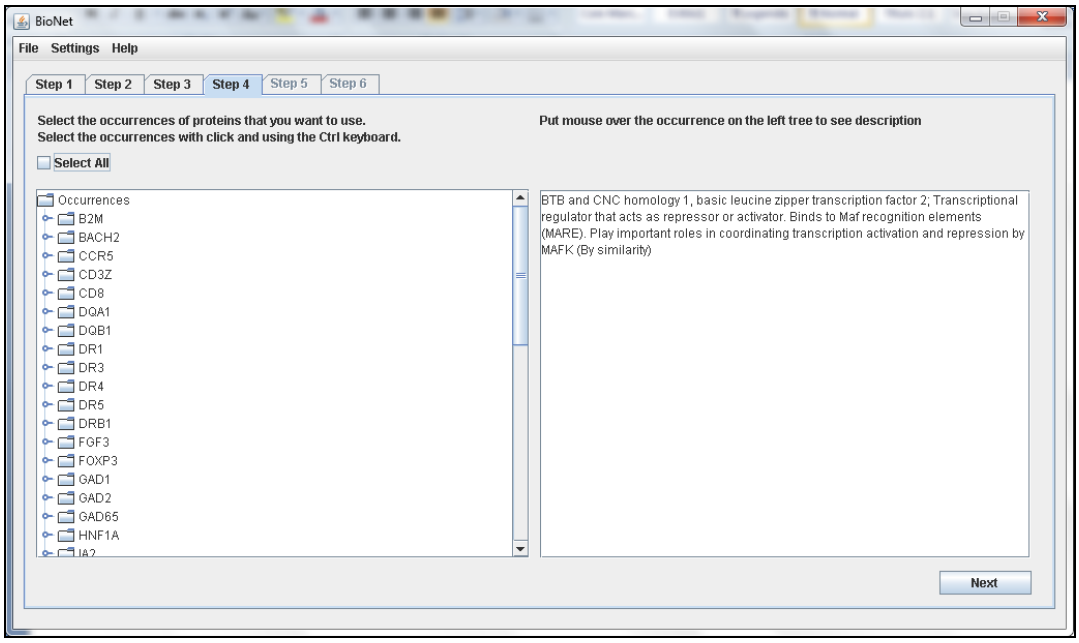

Figura 45 – Sistema *desktop* – *Step* 4

Ao clicar no botão *Next* o sistema verificará se existe uma árvore de ocorrências criada (linha 673) e chamará o método *processOccurrencesSelected* que retornará os nodos que

foram selecionados da árvore (linha 676). Com base na lista de nodos selecionados será executado o método *getOccurencesByTerm* que adicionará as ocorrências encontradas anteriormente ao nodo selecionado (linha 677).

O próximo método chamado *getPathProteinIdentification* executará a url de consulta ao site do STRING passando como parâmetro todas as ocorrências que foram selecionadas (Figura 46) e retornará a url da imagem da rede (linha 678). A seguir o próximo método a ser executado é o *getPathProteinImageAndXml* (linha 679) que receberá como parâmetro a url e através da expressão regular apresentada na Figura 48 separará a identificação da rede e montará a url de identificação, imagem da rede, arquivo XML e o arquivo texto. O método *setLinksToDownload* setará os links de acesso dos arquivos para *download* e vinculará aos eventos dos botões de *download*, conforme mostra a Figura 46.

```
671     btnStep4.addActionListener(new ActionListener() {
672         public void actionPerformed(ActionEvent arg0) {
673             if (jTreeOccurences != null) {
674                 startProcessMouse();
675                 TreeVO tree = new TreeVO(jTreeOccurences);
676                 ArrayList<String> listSelected = occurrenceBo.processOccurrencesSelected(tree);
677                 occurrencesSelected = stringBo.getOccurencesByTerm(listSelected, listOccurrences);
678                 linkIdentification = occurrenceBo.getPathProteinIdentification(occurrencesSelected);
679                 links = occurrenceBo.getPathProteinImageAndXml(linkIdentification);
680                 setLinksToDownload(links);
681                 enableTabbedPane(4, true);
682                 stopProcessMouse();
683             }
684         }
685     });
```

Figura 46 – Método de busca dos arquivos do STRING

```
http://string-db.org/api/url/networkList?identifiers=9606.ENSP00000296795%0A9606.ENSP00000368351%0A9606.ENSP00000258743%0A9606.ENSP00000288422%0A9606.ENSP00000401980&limit=0
```

Figura 47 – Exemplo da URL para consulta da rede selecionada

```
(e\\_([A-Za-z0-9]{2}[A-Za-z0-9\\_]+)\\.)
```

Figura 48 – Expressão regular para encontrar o identificador da rede

http://string-db.org/newstring_cgi/show_network_save_page.pl?taskId=99mnQqX1LX2a

Figura 49 – Exemplo da URL de consulta da rede de proteínas

4.5 Visualização da rede de interação da(s) proteína(s)

Na quinta aba da aplicação (Step 5) será disponibilizado ao usuário o *download* do arquivo XML, a imagem e o arquivo texto da rede, conforme mostra a Figura 50. Abaixo serão apresentados exemplos da url de busca imagem da rede de proteínas, url do arquivo XML e url do arquivo texto conforme mostrados na Figura 51, 52 e 53.

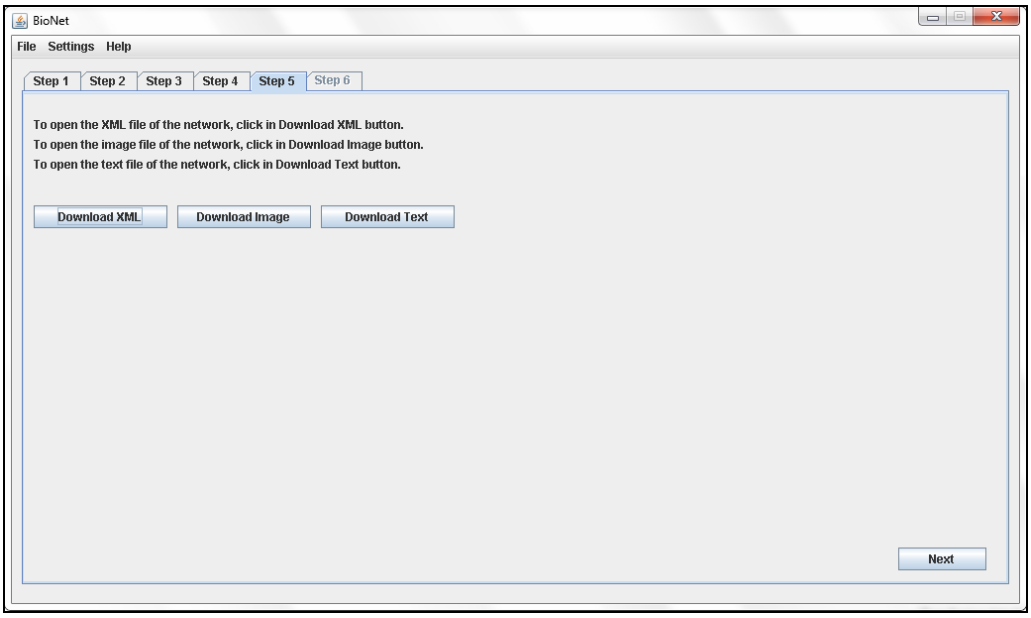

Figura 50 – Sistema *desktop* – *Step* 5

http://string-db.org/newstring_userdata/net_image_e_99mnQqX1LX2a_dw.png

Figura 51 – Exemplo da URL da imagem da rede de proteínas

http://string-db.org/newstring_userdata/xml_summary.99mnQqX1LX2a.xml

Figura 52 – Exemplo da URL de consulta do arquivo XML da rede de proteínas

http://string-db.org/newstring_userdata/proteins_desc.99mnQqX1LX2a.txt

Figura 53 – Exemplo da URL de consulta do arquivo TXT da rede de proteínas

4.6 Consulta alinhamento da rede de proteínas

Na sexta aba da aplicação (Step 6) serão apresentadas as proteínas selecionadas nos passos anteriores e, ao clicar no botão *Next* para cada uma das proteínas selecionadas em tela, será executado a url da Figura 55 passando como parâmetro a proteína e o nome da espécie. A url retornará um arquivo XML que será processado e o primeiro Id encontrado no arquivo XML será utilizado na consulta da próxima url, conforme Figura 56, que buscará o arquivo FASTA e novamente será processado para pegar a seqüência da proteína e por fim executará a url de consulta ao *site* do PathBLAST, conforme mostra a Figura 57. O sistema abrirá o navegador padrão com a url gerada, conforme mostra a Figura 58 e clicando em *Results* o site apresentará o resultado do processamento conforme mostra a Figura 59.

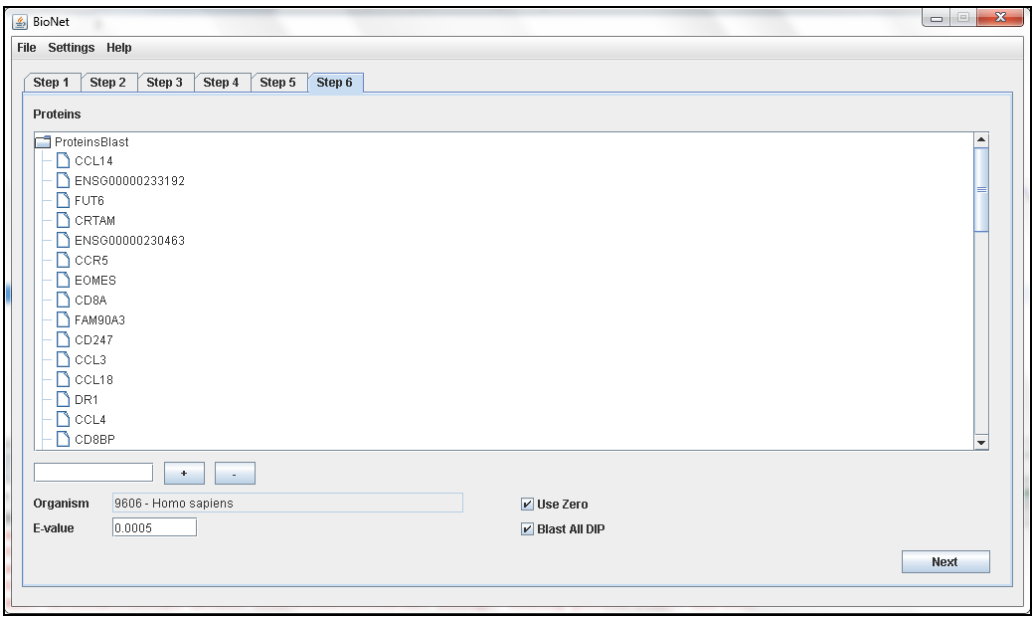

Figura 54 – Sistema *desktop* – *Step* 6

```
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=protein&term=CCL3L1+AND+Homo+AND+sapiens
```

Figura 55 – Exemplo da URL para buscar ID Primário

```
http://www.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=protein&id=27477072&complexity=0&rettype=fasta
```

Figura 56 – Exemplo da URL para buscar o arquivo FASTA

http://www.pathblast.org/blastpathway.jsp?A_id=CCL3L1_HUMAN&A_seq=MQVSTAAL
AVLLCTMALCNQVLSAPLAADTPTACCFSYTSRQIPQNFIADYFETSSQCSKPSVIFLT
KRGRQVCADPSEEWVQKYVSDLELSA&B_id=CCL14_HUMAN&B_seq=MKISVAAIP
FFLLITIALGTKTESSSRGPYHPSECCFTYTTYKIPRQRIMDYYETNSQCSKPGIVFITKR
GHSVCTNPSDKWVQDYIKDMKEN&C_id=B2M_HUMAN&C_seq=MSRSVALAVLAL
LSLSGLEAIQRTPKIQVYSRHPAENGKSNFLNCYVSGFHPSDIEVDLLKNGERIEKVEH
SDLSFSKDWSFYLLYYTEFTPTEKDEYACRVNHVTLSQPKIVKWDRDI&T_ORG=Ho
mo_sapiens.fa&E_VALUE=5.0E-4&useZero=true&blastAllDip=true&method=POST

Figura 57 – Exemplo da URL de consulta ao PathBlast

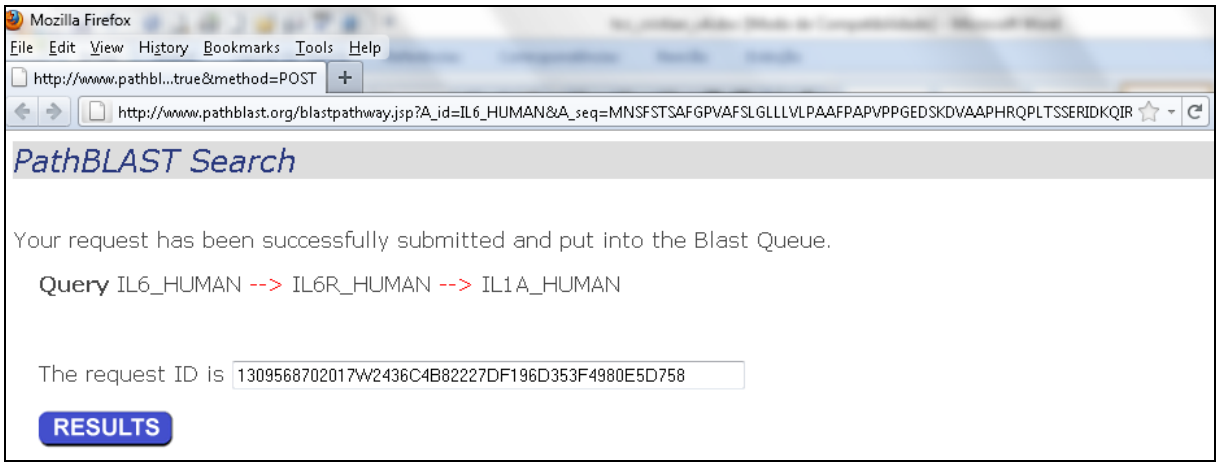

Figura 58 – Exemplo da URL de consulta ao alinhamento da rede no PathBLAST

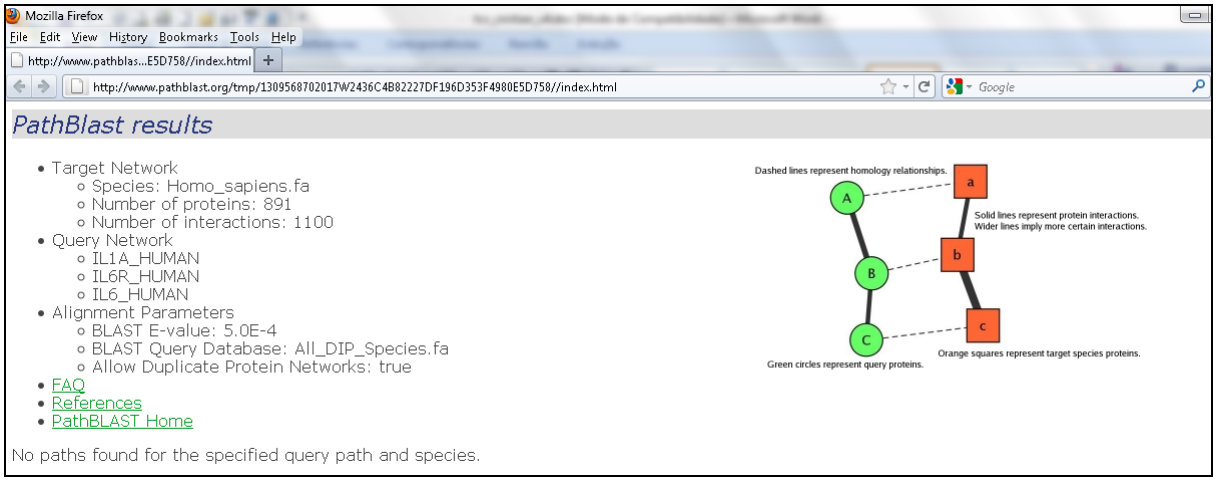

Figura 59 – Exemplo de resultado do alinhamento da rede no PathBLAST

4.7 Considerações finais

Nesse capítulo foram apresentados os principais métodos implementados no sistema e que serão executados na aplicação com as chamadas das urls e *web services* de consulta ao *site* OMIM, STRING e PathBLAST.

5 CENÁRIOS DE TESTES DO BIONET 2.0

Nas subseções que seguem serão apresentados os cinco cenários de testes que foram realizados no sistema *desktop*.

5.1 Primeiro cenário de testes

Iniciamos a pesquisa procurando pelo termo “alergics”, como mostra a Figura 60, recebendo como resposta que o item não foi encontrado, como mostra a Figura 61, na seqüência procurou pelos termos “sarampo”, “cachumba”, e “caxumba” obtendo a mesma resposta, já que esse quatro termos não existem na base de dados do OMIM.

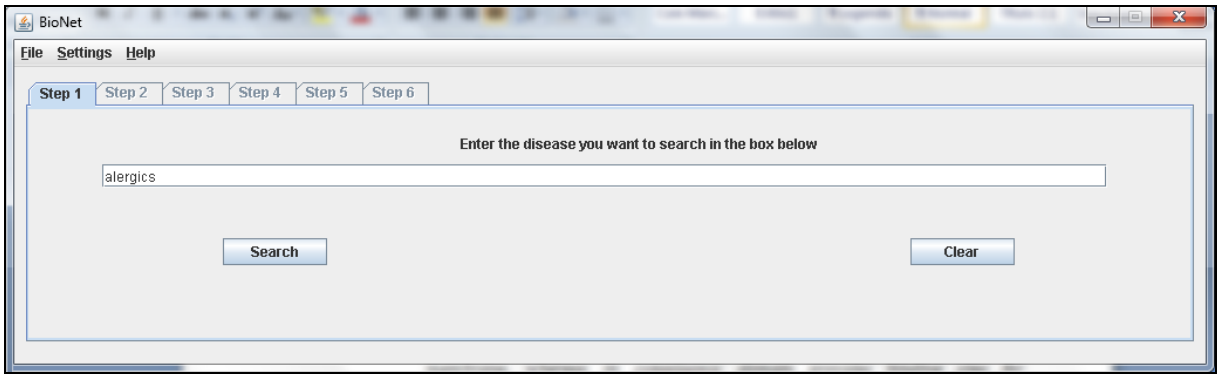

Figura 60 – Pesquisa inicial do termo “alergics”

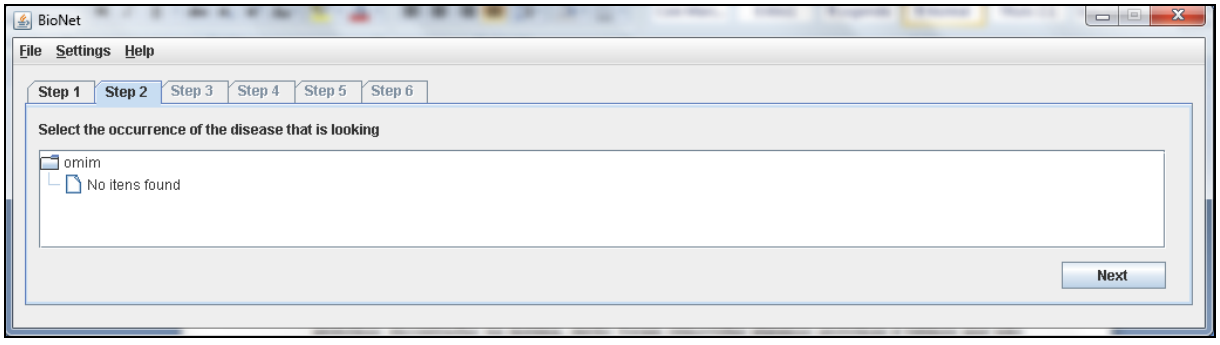

Figura 61 – Resultado pesquisa de item não encontrado

Então procurou pelo termo “malaria”, obtendo assim a lista de ocorrências de doenças com esse termo conforme mostra a Figura 63. Selecionou a doença com o identificador “+109270” e com descrição principal “SOLUTE CARRIR FAMILY 4 (ANION EXCHANGER), MEMBER 1; SLC4A1” conforme mostra a Figura 64.

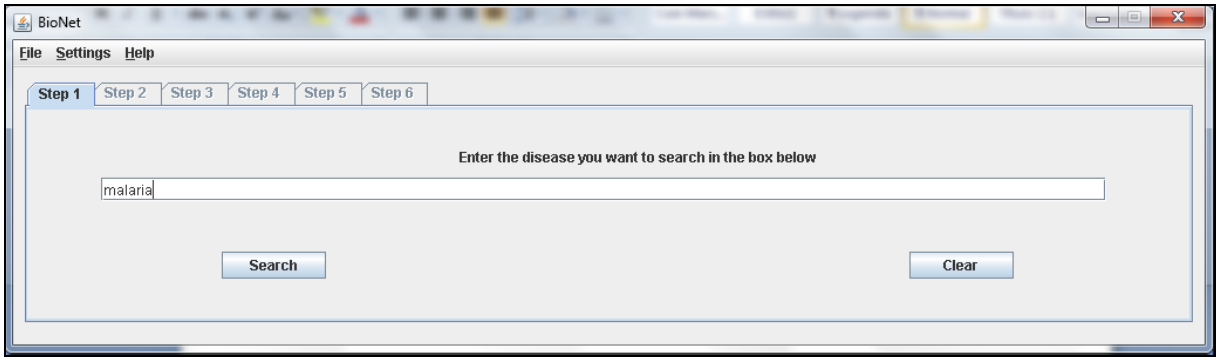

Figura 62 – Pesquisa do termo “malaria”

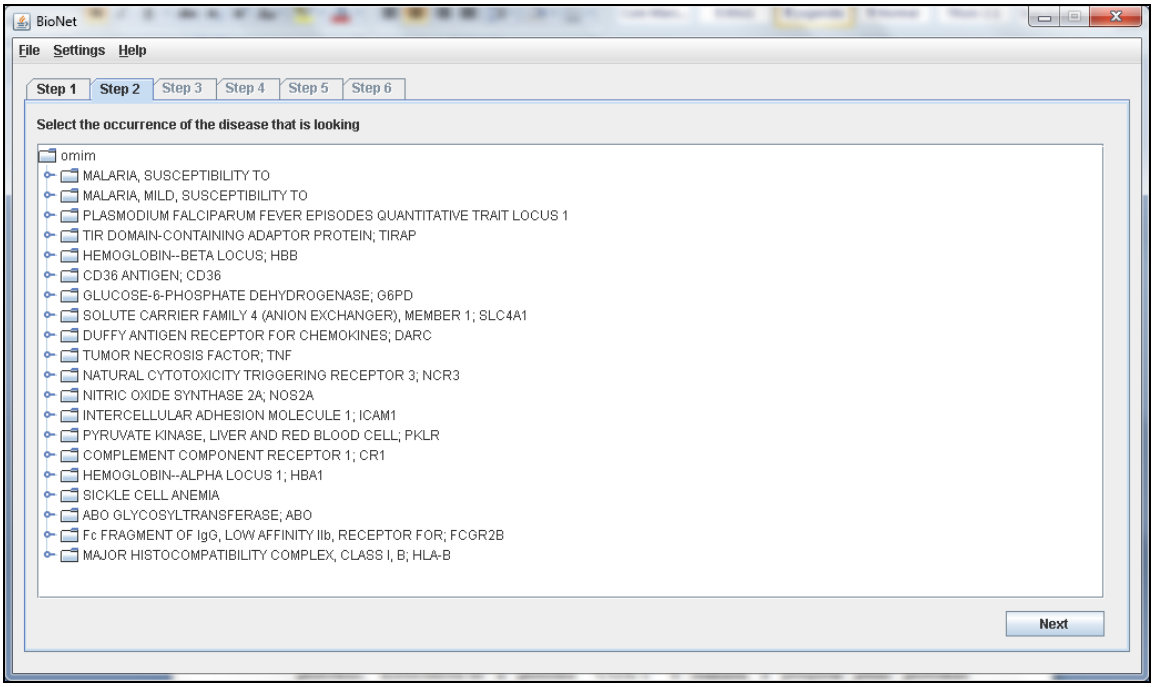

Figura 63 – Resultado pesquisa do termo “malaria”

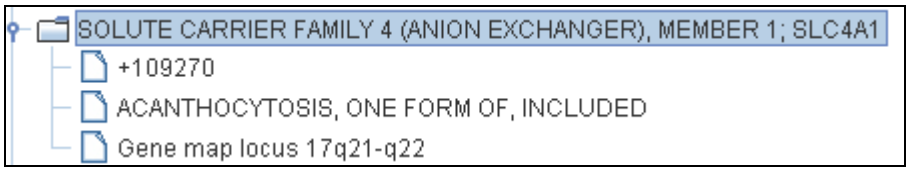

Figura 64 – Doença selecionada no primeiro cenário de testes

Após isso foram apresentadas algumas sugestões de proteínas encontradas para a doença selecionada conforme mostra a Figura 65.

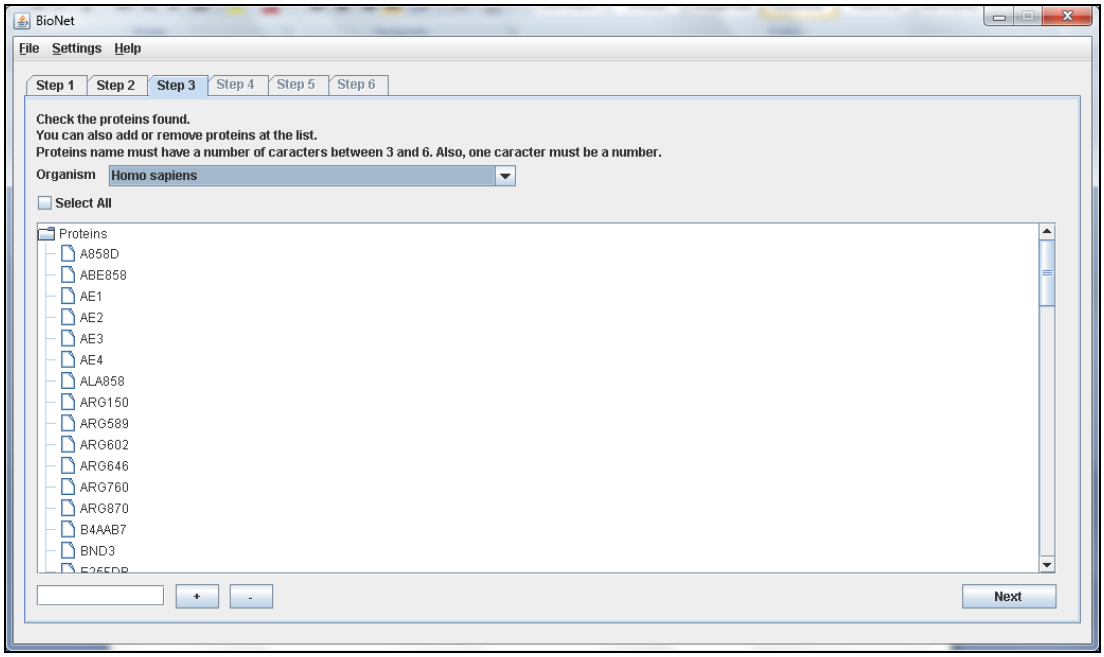

Figura 65 – Resultado pesquisa das proteínas

Então foram removidas algumas proteínas e termos que não eram proteínas, selecionou-se o organismo “Homo sapiens”, acrescentou-se a proteína “COX-1” e realizou a pesquisa pelas proteínas “SLC4A1”, “BND3”, “EMPB3”, “EPB3”, “AE1” e “COX-1” conforme Figura 66, obtendo assim a lista de ocorrências das proteínas em humanos.

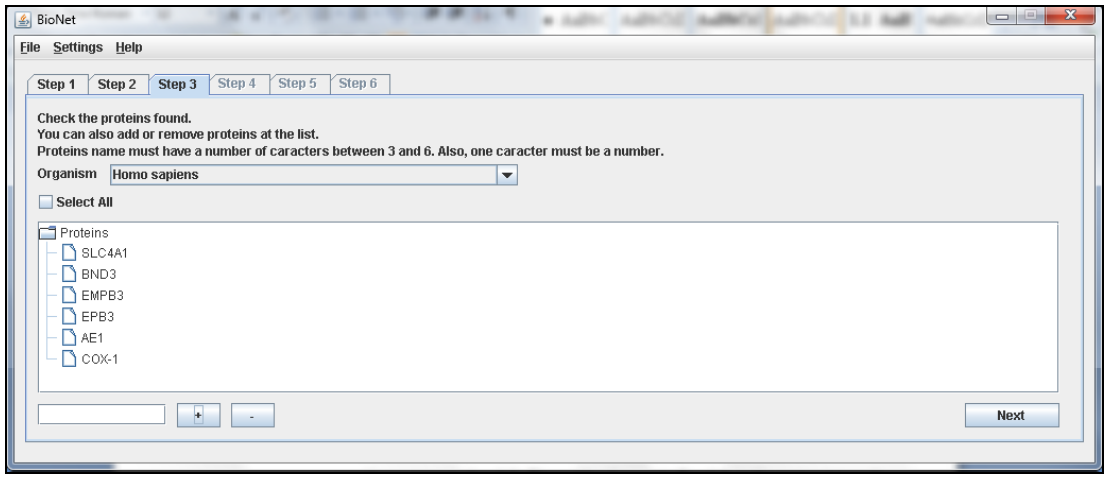

Figura 66 – Lista de proteínas que serão pesquisadas

Então selecionou a ocorrência da proteína “SLC4A1” e clicou para prosseguir conforme mostra a Figura 67, podendo assim visualizar a rede de interação da proteína, conforme mostra a Figura 68.

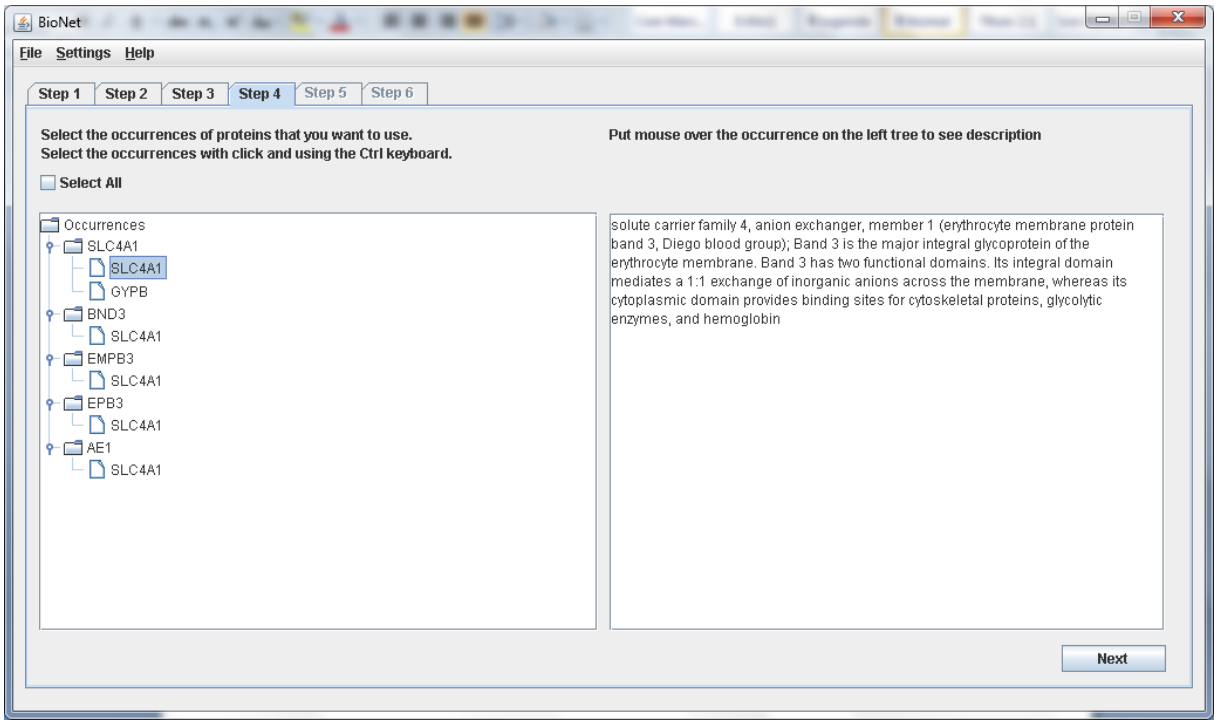

Figura 67 – Lista de ocorrências que serão pesquisadas

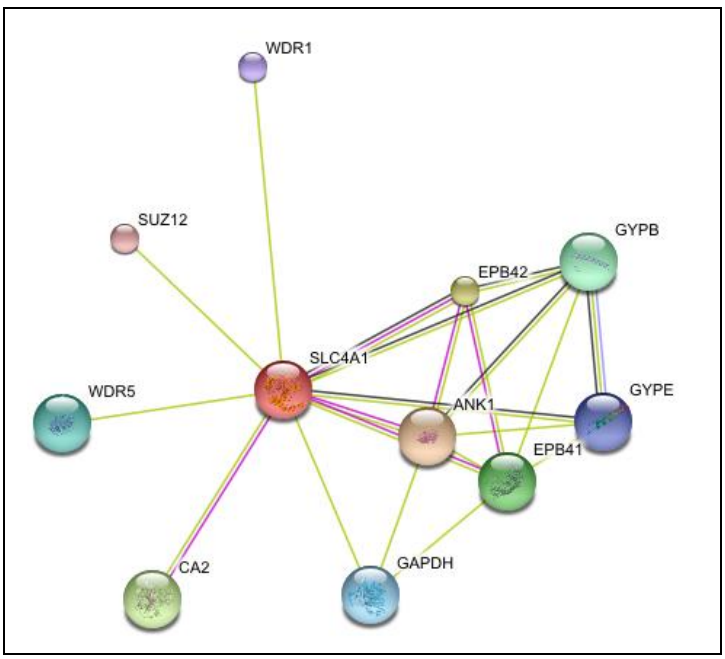

Figura 68 – Primeiro cenário de testes rede STRING

Após isso realizou o *download* do arquivo XML da rede de interação da proteína, conforme mostra a Figura 69 e importou o mesmo no *software* Cytoscape.

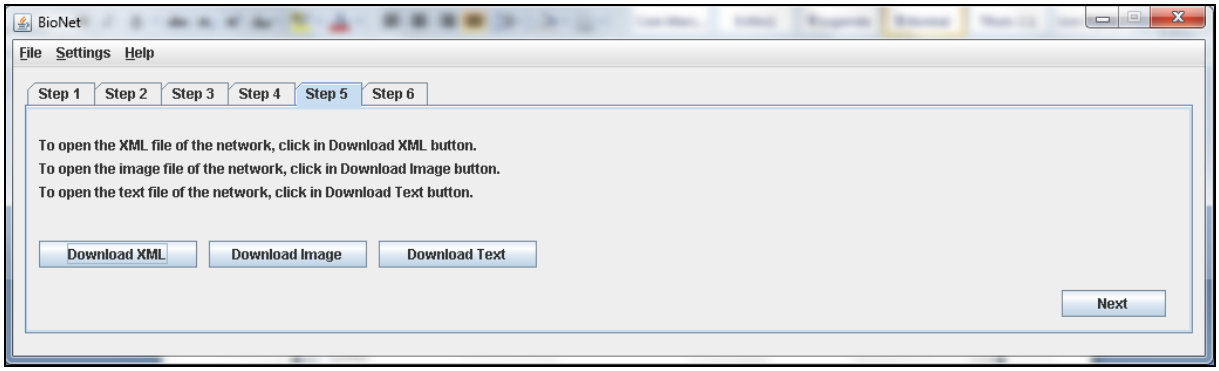

Figura 69 – *Download* do arquivo XML, imagem da rede e arquivo texto

No próximo passo selecionou-se as proteínas “EPB42”, “SLC4A1” e “WDR5” clicou para prosseguir conforme mostra a Figura 70.

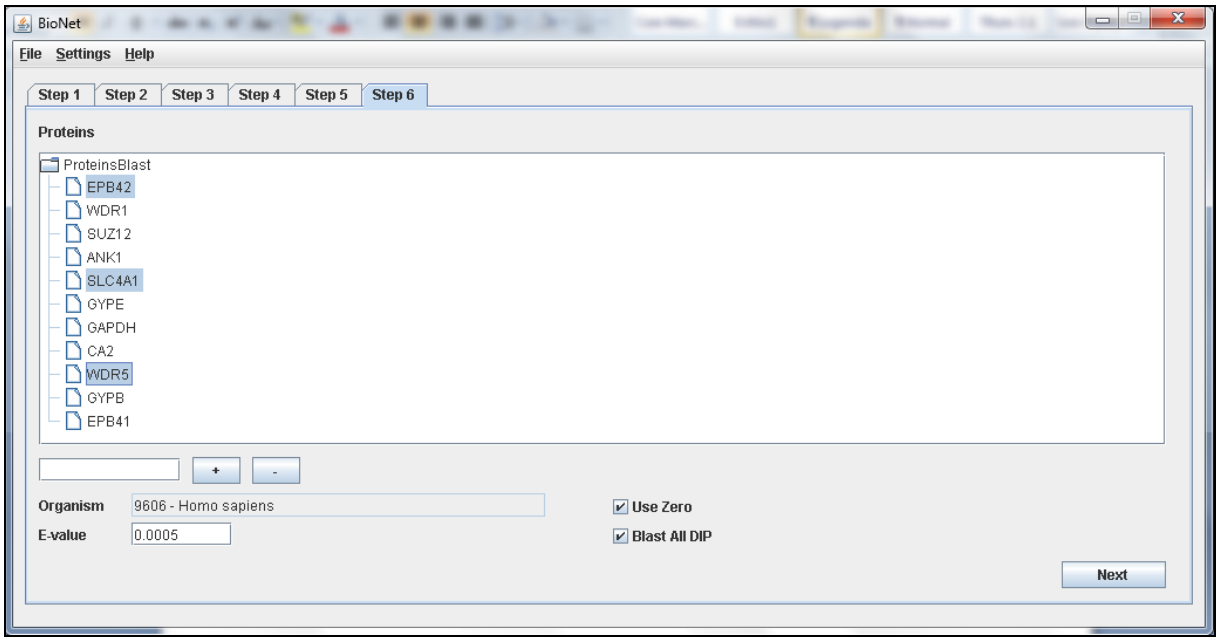

Figura 70 – Seleção das proteínas para pesquisa no PathBLAST

A consulta ao site do PathBLAST foi realizada apresentando a Figura 71 e o resultado do alinhamento das proteínas selecionadas foi apresentado conforme mostra a Figura 72.

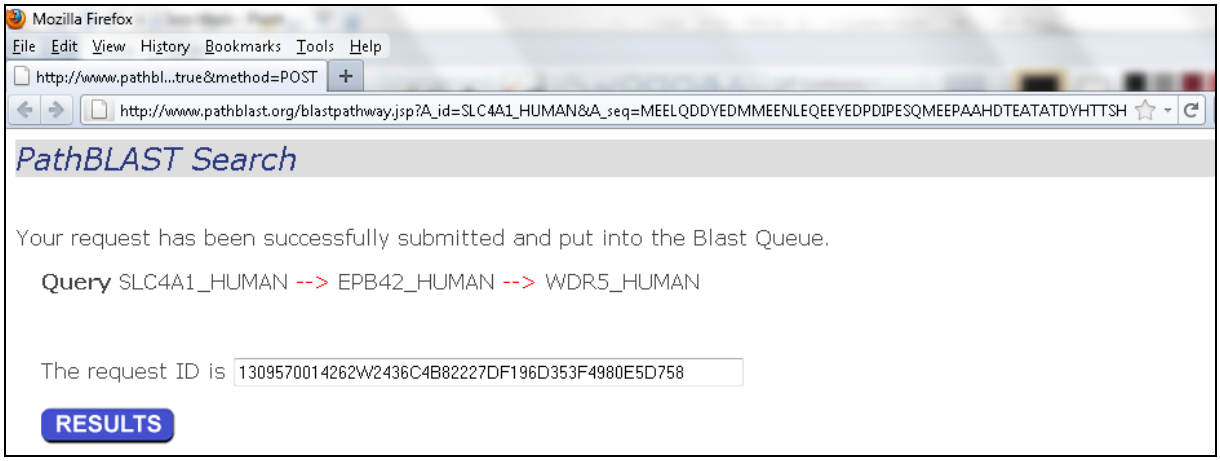

Figura 71 – Consulta das proteínas ao site do PathBLAST

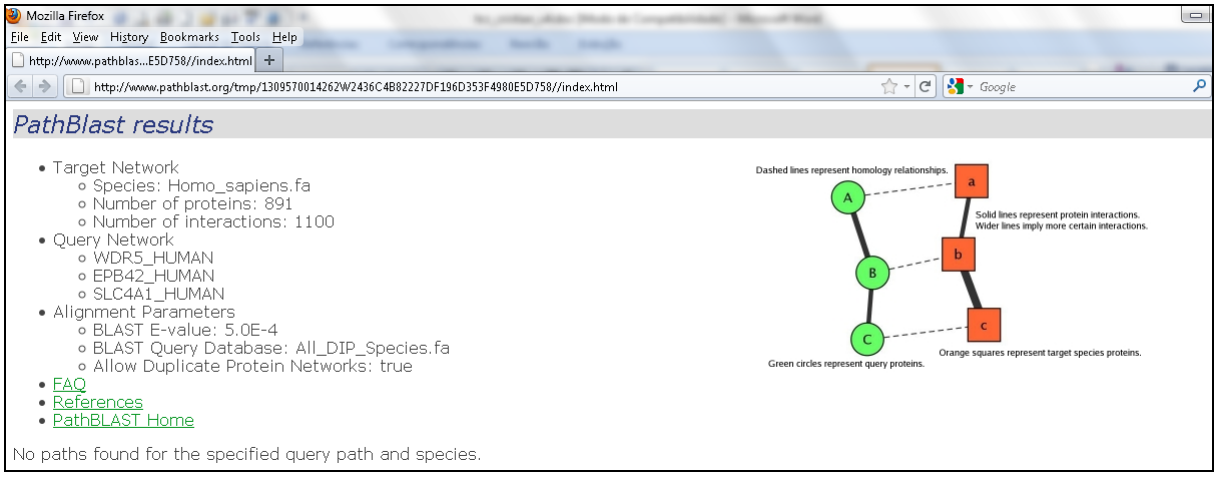

Figura 72 – Resultado da consulta ao site do PathBLAST

5.2 Segundo cenário de testes

No segundo cenário, iniciou a pesquisa procurando pelo termo “huntington”, obtendo assim a lista de ocorrências de doenças com esse termo. Selecionou a doença com o identificador “#143100” e com descrição principal “HUNTINGSTON DISEASE; HD”, conforme mostra a Figura 73.

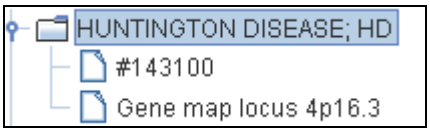

Figura 73 – Doença selecionada no segundo cenário de testes

Após isso foram apresentadas algumas sugestões de proteínas encontradas para a doença selecionada conforme mostra a Figura 74.

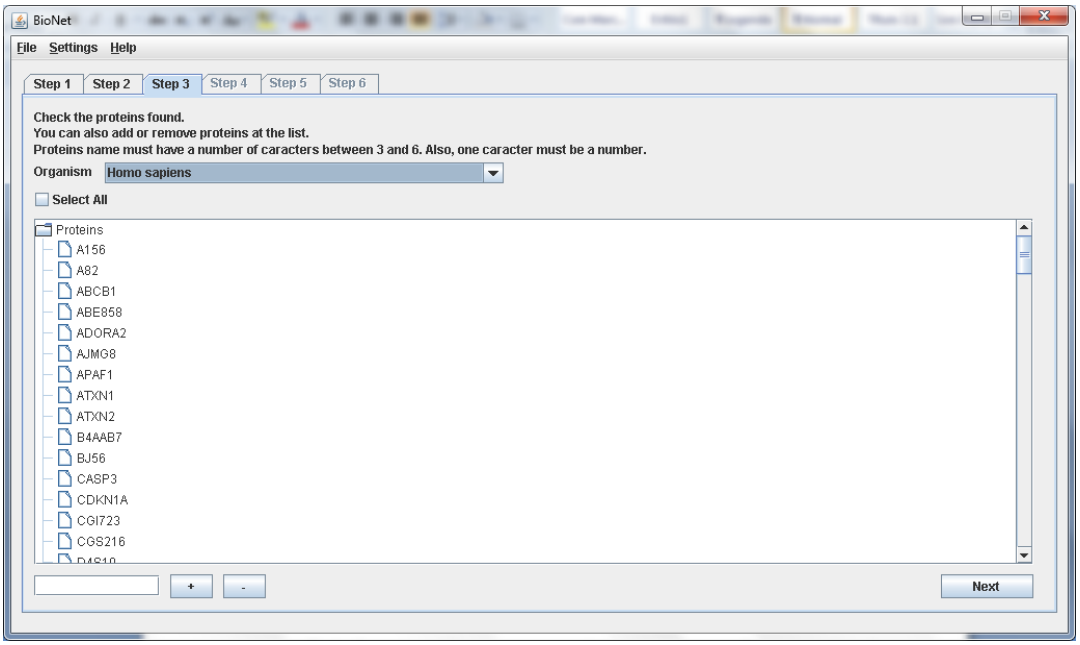

Figura 74 – Lista de sugestões de proteínas que serão pesquisadas

Selecionou-se o organismo “Homo sapiens”, foram removidos os termos que não eram proteínas e realizou a pesquisa pelas proteínas “HD”, “HTT”, “MRI”, “XL”, “DM1”, “NF1”, “NF2”, “D4S10”, “SE”, “G8” e “MMSE” conforme mostra a Figura 75.

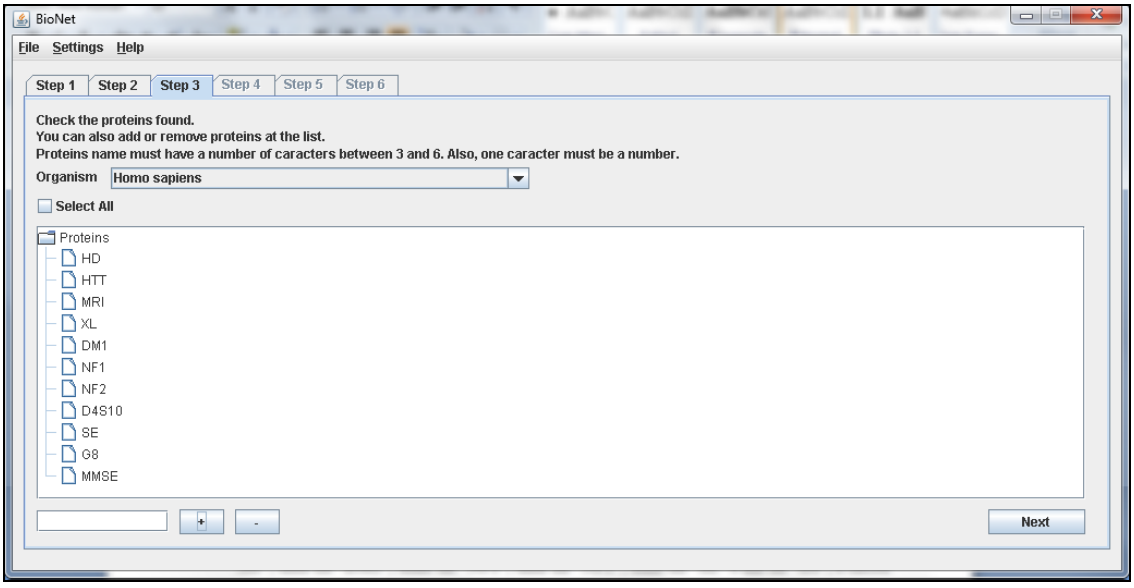

Figura 75 – Lista de proteínas que serão pesquisadas

Obteve-se assim a lista de ocorrências das proteínas em humanos conforme mostra a Figura 76.

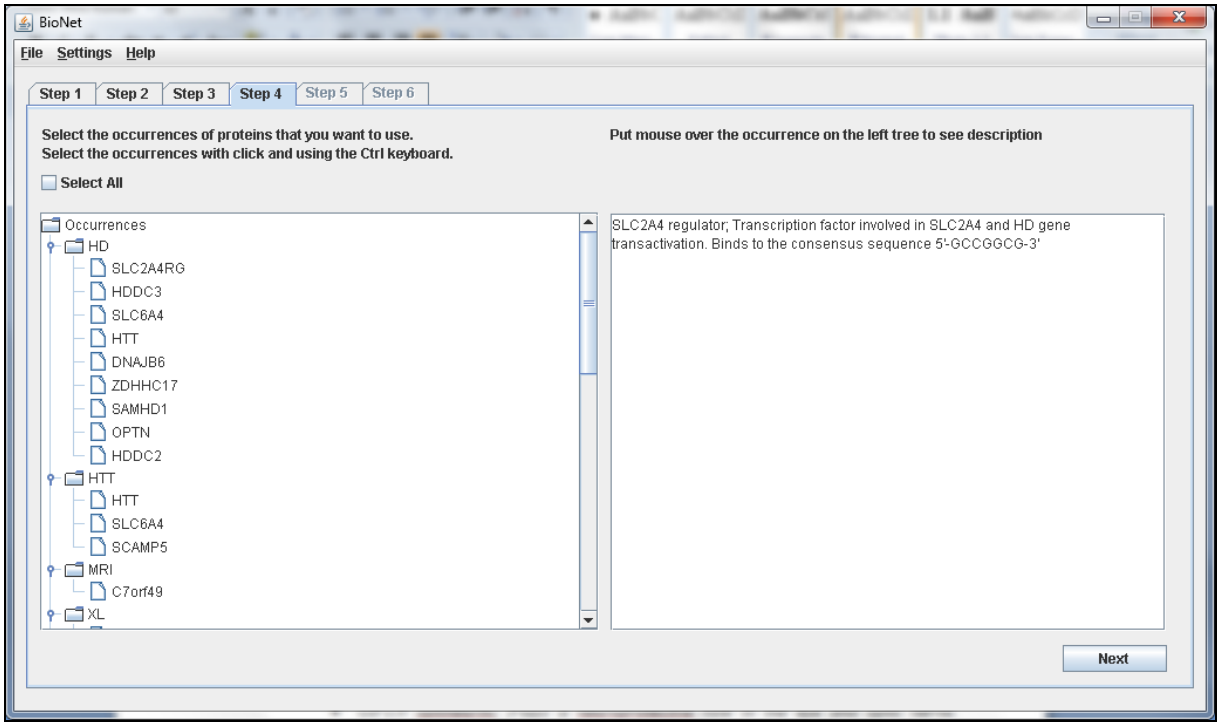

Figura 76 – Lista de ocorrências que serão pesquisadas

Então selecionou-se quatorze ocorrências de proteínas, quatro da "HD", uma da "HTT", uma da "MRI", uma da "XL", uma da "DM1", duas da "NF1", uma da "NF2", duas da "SE" e um da "G8", e clicou para prosseguir, podendo assim visualizar a rede de interação das proteínas, conforme mostra a Figura 77. Abaixo as descrições das quatorze proteínas selecionadas:

- *OPTN optineurin; Plays a neuroprotective role in the eye and optic nerve. Probably part of the TNF-alpha signaling pathway that can shift the equilibrium toward induction of cell death. May act by regulating membrane trafficking and cellular morphogenesis via a complex that contains Rab8 and hungtingtin (HD). May constitute a cellular target for adenovirus E3 14.7, an inhibitor of TNF-alpha functions, thereby affecting cell death*;

- *HDDC3 HD domain containing 3*;

- *SAMHD1 SAM domain and HD domain 1; Putative nuclease involved in innate immune response by acting as a negative regulator of the cell-intrinsic antiviral response. May play a role in mediating proinflammatory responses to TNF-alpha signaling*;

- *SLC6A4 solute carrier family 6 (neurotransmitter transporter, serotonin), member 4; Terminates the action of serotonin by its high affinity sodium-dependent reuptake into presynaptic terminals;*

- *HTT huntingtin; May play a role in microtubule-mediated transport or vesicle function*;

- *C7orf49 Modulator of retrovirus infection homolog ; May act as a regulator of proteasome (By similarity)*;

- *RTN4 reticulon 4; Potent neurite growth inhibitor in vitro and plays a role both in the restriction of axonal regeneration after injury and in structural plasticity in the CNS. Isoform 2 reduces the anti-apoptotic activity of Bcl-xl and Bcl-2. This is likely consecutive to their change in subcellular location, from the mitochondria to the endoplasmic reticulum, after binding and sequestration. Isoform 2 and isoform 3 inhibit BACE1 activity and amyloid precursor protein processing*;

- *TCEA2 transcription elongation factor A (SII), 2; Necessary for efficient RNA polymerase II transcription elongation past template-encoded arresting sites. The arresting sites in DNA have the property of trapping a certain fraction of elongating RNA polymerases that pass through, resulting in locked ternary complexes. Cleavage of the nascent transcript by S-II allows the resumption of elongation from the new 3'-terminus;*

- *NF1L4 Putative neurofibromin 1-like protein 4/6 Precursor ; Stimulates the GTPase activity of Ras. NF1 shows greater affinity for Ras GAP, but lower specific activity. May be a regulator of Ras activity*;

- *APOBEC1 apolipoprotein B mRNA editing enzyme, catalytic polypeptide 1; Catalytic component of the apolipoprotein B mRNA editing enzyme complex which is responsible for the postranscriptional editing of a CAA codon for Gln to a UAA codon for stop in the APOB mRNA. Also involved in CGA (Arg) to UGA (Stop) editing in the NF1 mRN;*

- *SGSM3 small G protein signaling modulator 3; May play a cooperative role in NF2-mediated growth suppression of cells;*

- *FUT1 fucosyltransferase 1 (galactoside 2-alpha-L-fucosyltransferase, H blood group); Creates a soluble precursor oligosaccharide FuC-alpha ((1,2)Galbeta-) called the H antigen which is an essential substrate for the final step in the soluble A and B antigen synthesis pathway. H and Se enzymes fucosylate the same acceptor substrates but exhibit different Km values;*

- *FUT2 fucosyltransferase 2 (secretor status included); Creates a soluble precursor oligosaccharide FuC-alpha ((1,2)Galbeta-) called the H antigen which is an essential substrate for the final step in the soluble A and B antigen synthesis pathway. H and Se enzymes fucosylate the same acceptor substrates but exhibit different Km values;*

- *U52 small nucleolar RNA, C/D box 52;*

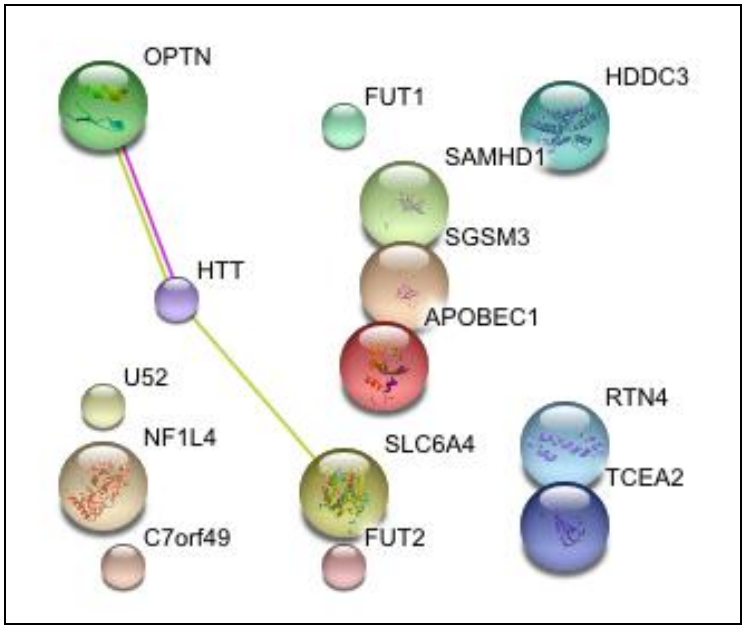

Figura 77 – Segundo cenário de testes rede STRING

Após isso realizou o *download* do arquivo XML da rede de interação da proteína, conforme mostra a Figura 78, e importou-se o mesmo no *software* Cytoscape.

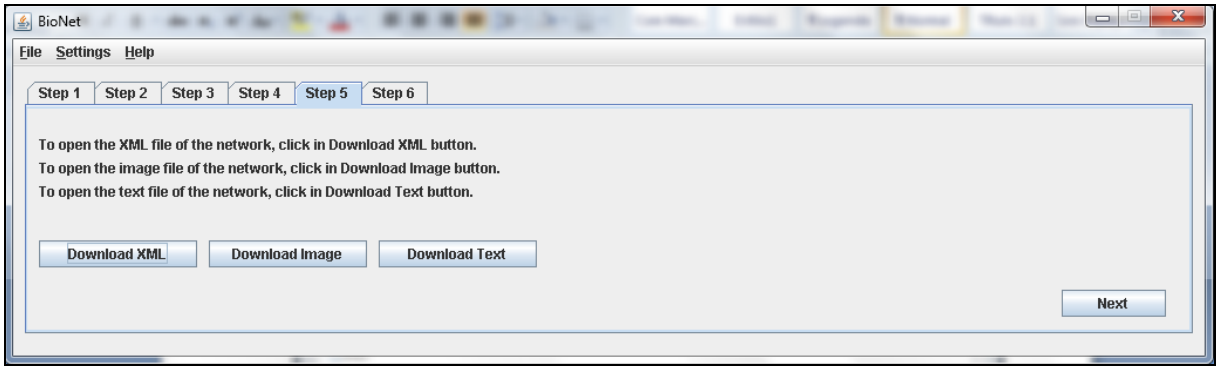

Figura 78 – *Download* do arquivo XML, imagem da rede e arquivo texto

No próximo passo selecionou-se as proteínas “FUT1”, “C7orf49” e “TCEA2”, e clicou-se para prosseguir conforme mostra a Figura 79.

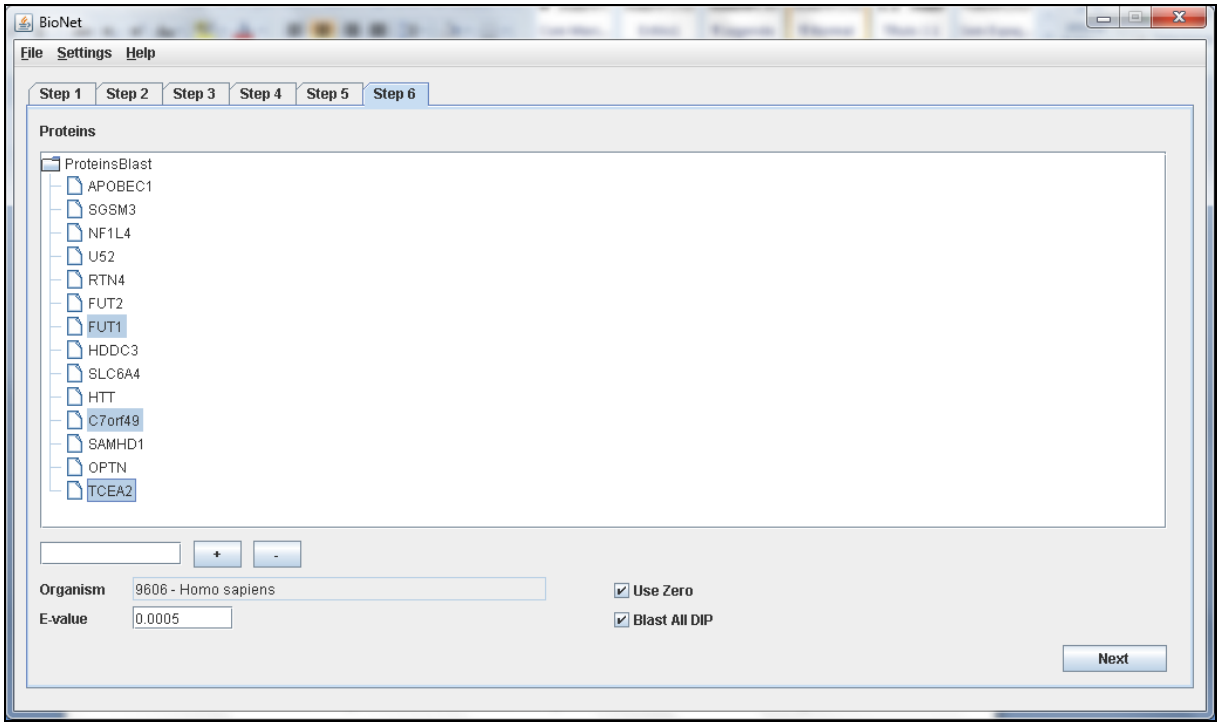

Figura 79 – Seleção das proteínas para pesquisa no PathBLAST

A consulta ao site do PathBLAST foi realizada apresentando a Figura 80 e o resultado do alinhamento das proteínas selecionadas foi apresentado conforme mostra a Figura 81.

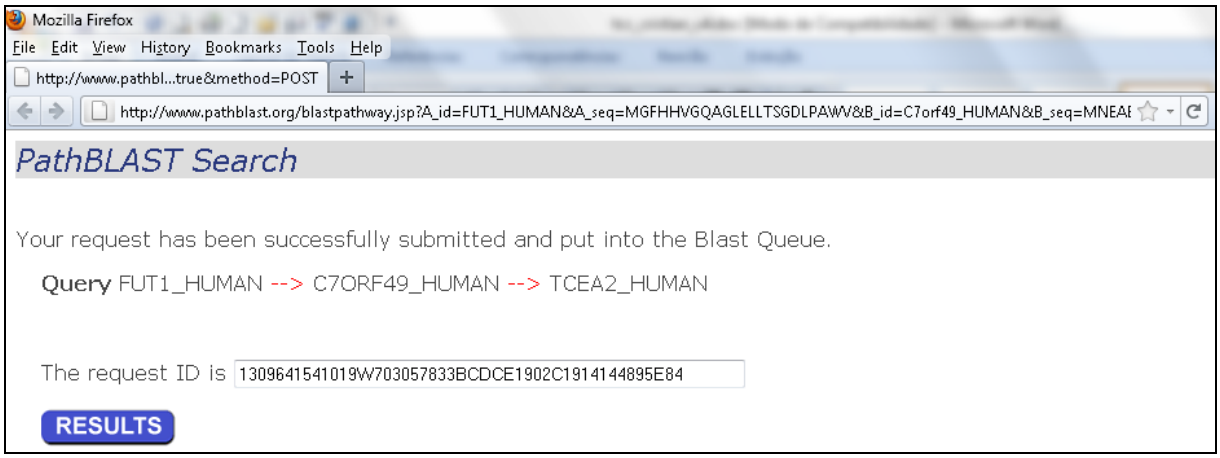

Figura 80 – Consulta das proteínas ao site do PathBLAST

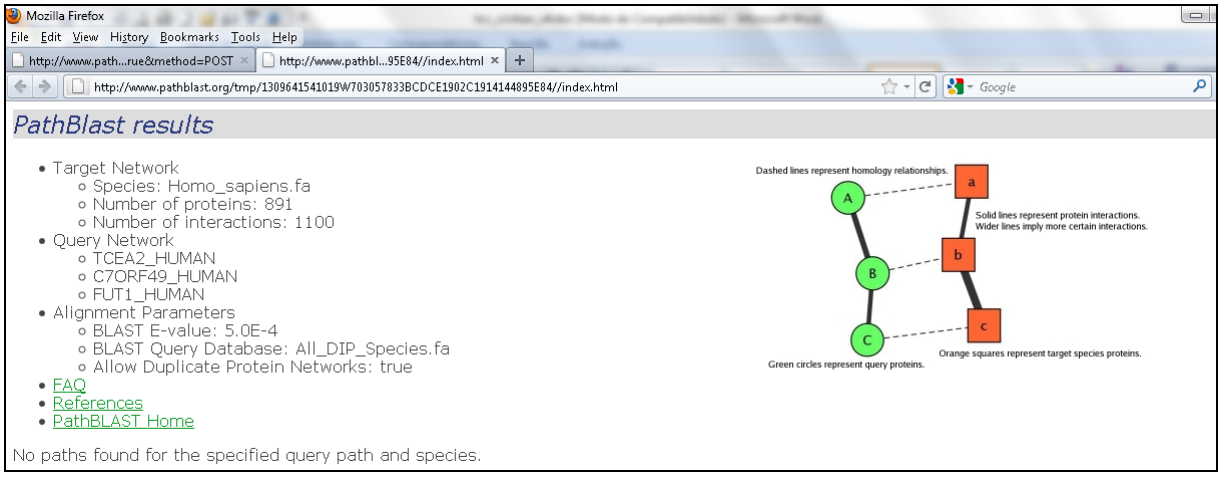

Figura 81 – Resultado da consulta ao site do PathBLAST

5.3 Terceiro cenário de testes

No terceiro cenário, iniciou a pesquisa procurando pelo termo “charcot marie”, obtendo assim a lista de ocorrências de doenças com esse termo. Selecionou a doença com o identificador “#604563” e com descrição principal “CHARCOT-MARIE-TOOTH DISEASE, TYPE 4B2, CMT4B2” conforme mostra a Figura 82.

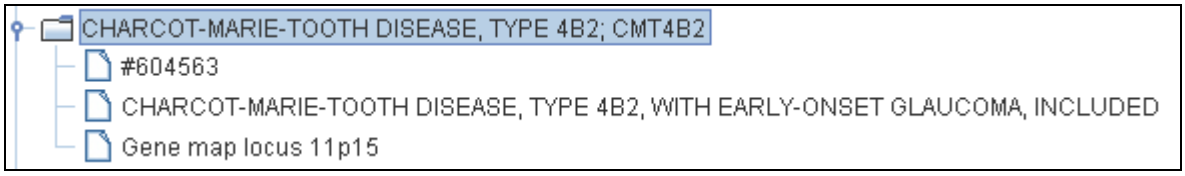

Figura 82 – Doença selecionada no terceiro cenário de testes

Após isso foram apresentadas algumas sugestões de proteínas encontradas para a doença selecionada conforme mostra a Figura 83.

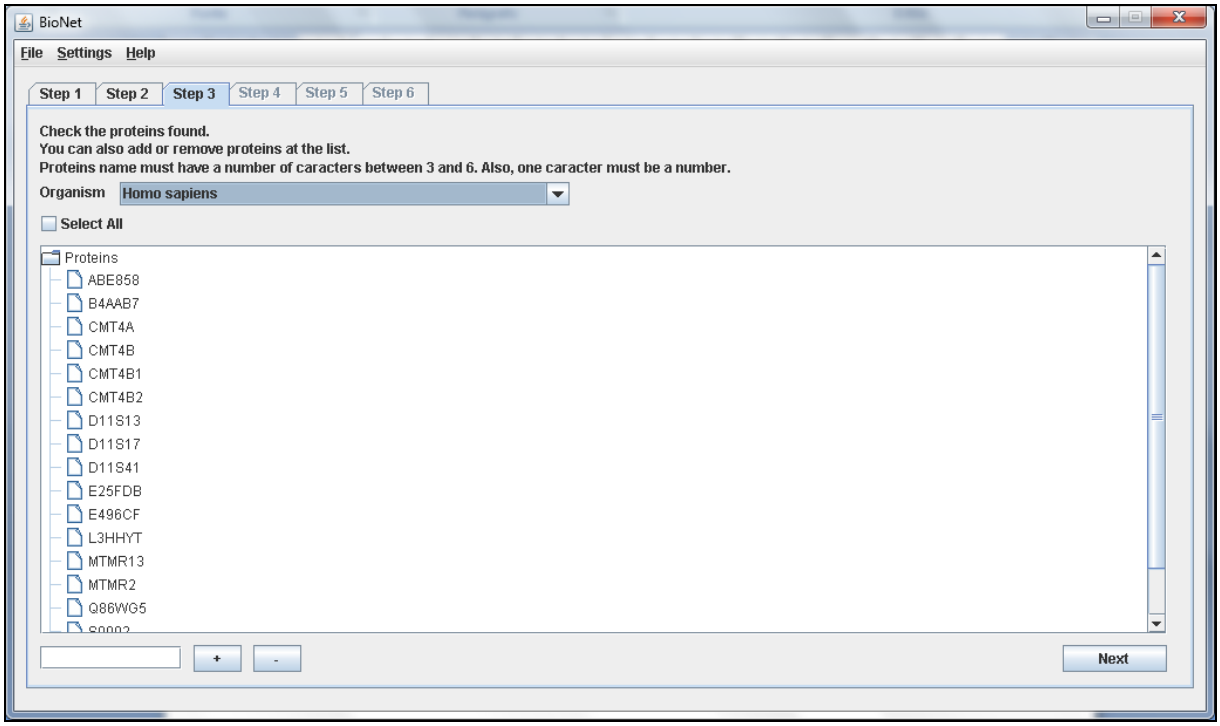

Figura 83 – Lista de sugestões de proteínas que serão pesquisadas

Selecionou-se o organismo “Homo sapiens”. Foram removidos os termos que não eram proteínas e realizou a pesquisa pelas proteínas “CMT4B2”, “SBF2”, “CMT4B1”, “MTMR2”, “CMT”, “CMT4A”, “CSF”, “CMT4B” e “MTMR13” conforme mostra a Figura 84.

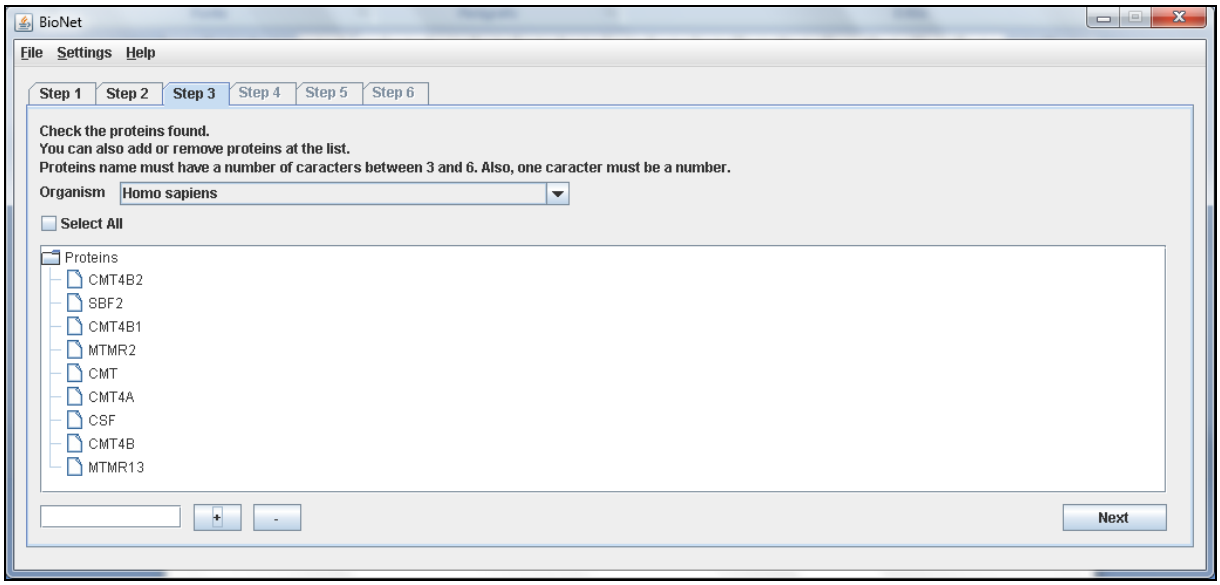

Figura 84 – Lista de proteínas que serão pesquisadas

Obteve-se assim a lista de ocorrências das proteínas em humanos, conforme mostra a Figura 85.

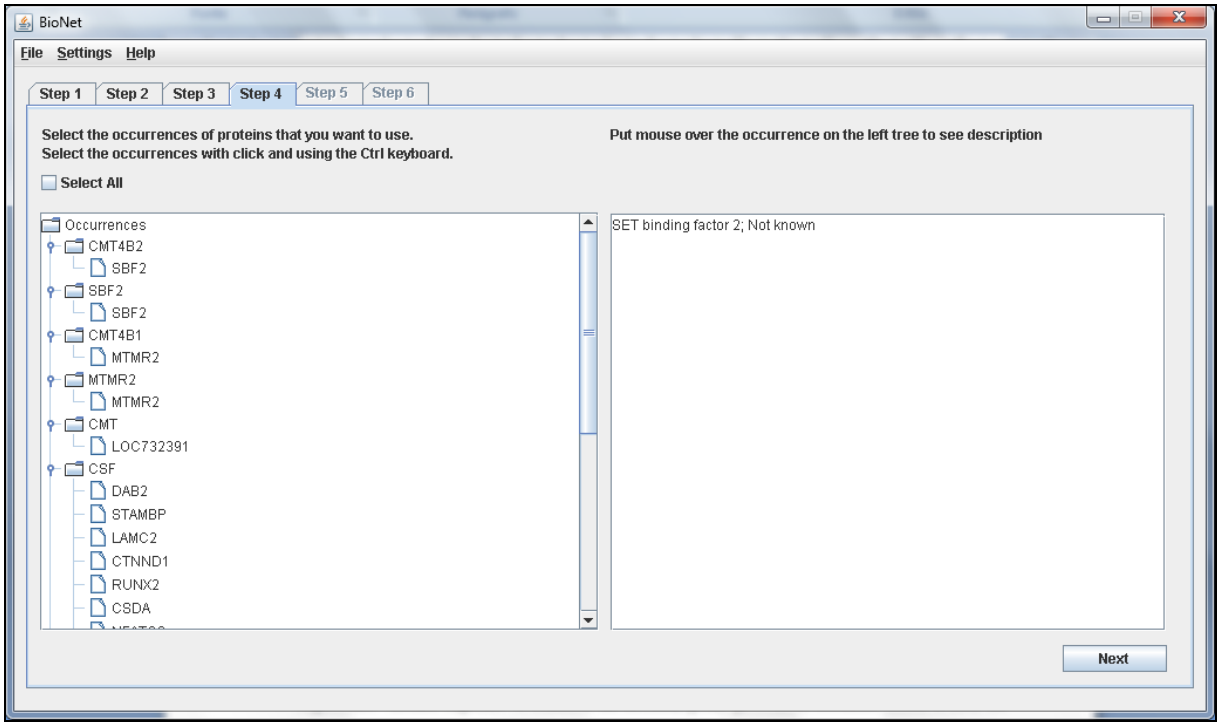

Figura 85 – Lista de ocorrências que serão pesquisadas

Então selecionou-se quatorze ocorrências de proteínas, uma da "CMT4B2", uma da "SBF2", uma da "CMT4B1", uma da "MTMR2", uma da "CMT", sete da "CSF", uma da "CMT4B" e uma da "MTMR13", e clicou-se para prosseguir, podendo-se assim visualizar a rede de interação da proteína, conforme mostra a Figura 86. Abaixo as descrições das proteínas selecionadas, sendo que as repetidas não serão citadas mais de uma vez:

- *SBF2 SET binding factor 2; Not known;*

- *MTMR2 myotubularin related protein 2; Phosphatase that acts on lipids with a phosphoinositol headgroup. Has phosphatase activity towards phosphatidylinositol- 3-phosphate and phosphatidylinositol-3,5-bisphosphate*

- *LOC732391 CMT duplicated region transcript 1;*

- *DAB2 disabled homolog 2, mitogen-responsive phosphoprotein (Drosophila); Component of the CSF-1 signal transduction pathway (By similarity);*

- *LAMC2 laminin, gamma 2; Binding to cells via a high affinity receptor, laminin is thought to mediate the attachment, migration and organization of cells into tissues during embryonic development by interacting with other extracellular matrix components. Ladsin exerts cell- scattering activity toward a wide variety of cells, including epithelial, endothelial, and fibroblastic cells;*

- *CSDA cold shock domain protein A pseudogene 1; Binds to the GM-CSF promoter. Seems to act as a repressor. Binds also to full length mRNA and to short RNA sequences containing the consensus site 5'-UCCAUCA-3'. May have a role in translation repression (By similarity);*

- *OSM oncostatin M; Growth regulator. Inhibits the proliferation of a number of tumor cell lines. Stimulates proliferation of AIDS-KS cells. It regulates cytokine production, including IL-6, G-CSF and GM-CSF from endothelial cells. Uses both type I OSM receptor (heterodimers composed of LIPR and IL6ST) and type II OSM receptor (heterodimers composed of OSMR and IL6ST);*

- *IL10 interleukin 10; Inhibits the synthesis of a number of cytokines, including IFN-gamma, IL-2, IL-3, TNF and GM-CSF produced by activated macrophages and by helper T-cells;*

- *CBFB core-binding factor, beta subunit; CBF binds to the core site, 5'-PYGPYGGT-3', of a number of enhancers and promoters, including murine leukemia virus, polyomavirus enhancer, T-cell receptor enhancers, LCK, IL-3 and GM-CSF promoters. CBFB enhances DNA binding by RUNX1;*

- *CSF1 colony stimulating factor 1 (macrophage); Granulocyte/macrophage colony-stimulating factors are cytokines that act in hematopoiesis by*

controlling the production, differentiation, and function of 2 related white cell populations of the blood, the granulocytes and the monocytes-macrophages. CSF- 1 induces cells of the monocyte/macrophage lineage. It plays a role in immunological defenses, bone metabolism, lipoproteins clearance, fertility and pregnancy;

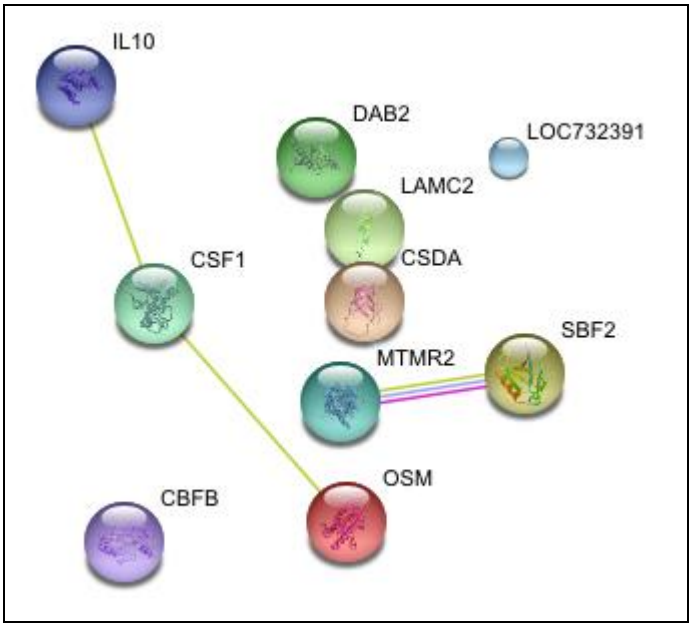

Figura 86 – Terceiro cenário de testes rede STRING

Após isso realizou-se o *download* do arquivo XML da rede de interação da proteína, conforme mostra a Figura 87 e importou-se o mesmo no *software* Cytoscape.

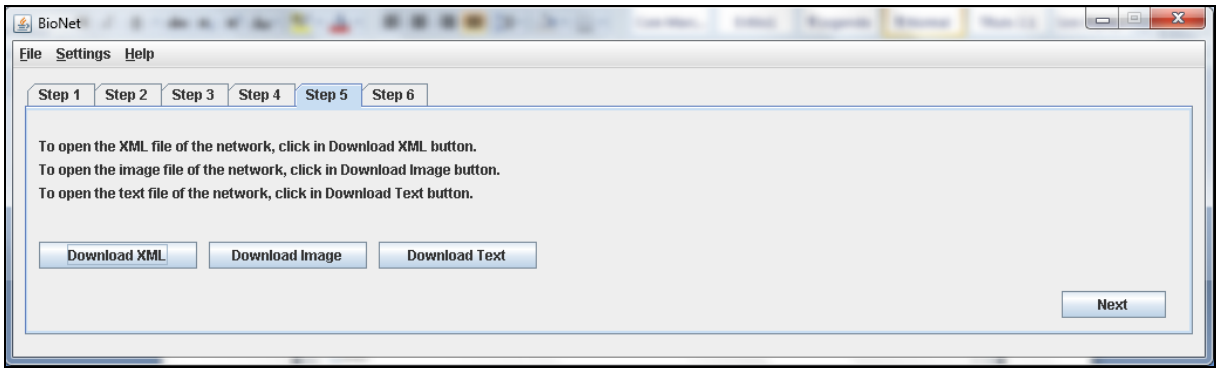

Figura 87 – *Download* do arquivo XML, imagem da rede e arquivo texto

No próximo passo selecionou-se as proteínas “SBF2”, “OSM” e “IL10” e clicou-se para prosseguir conforme mostra a Figura 88.

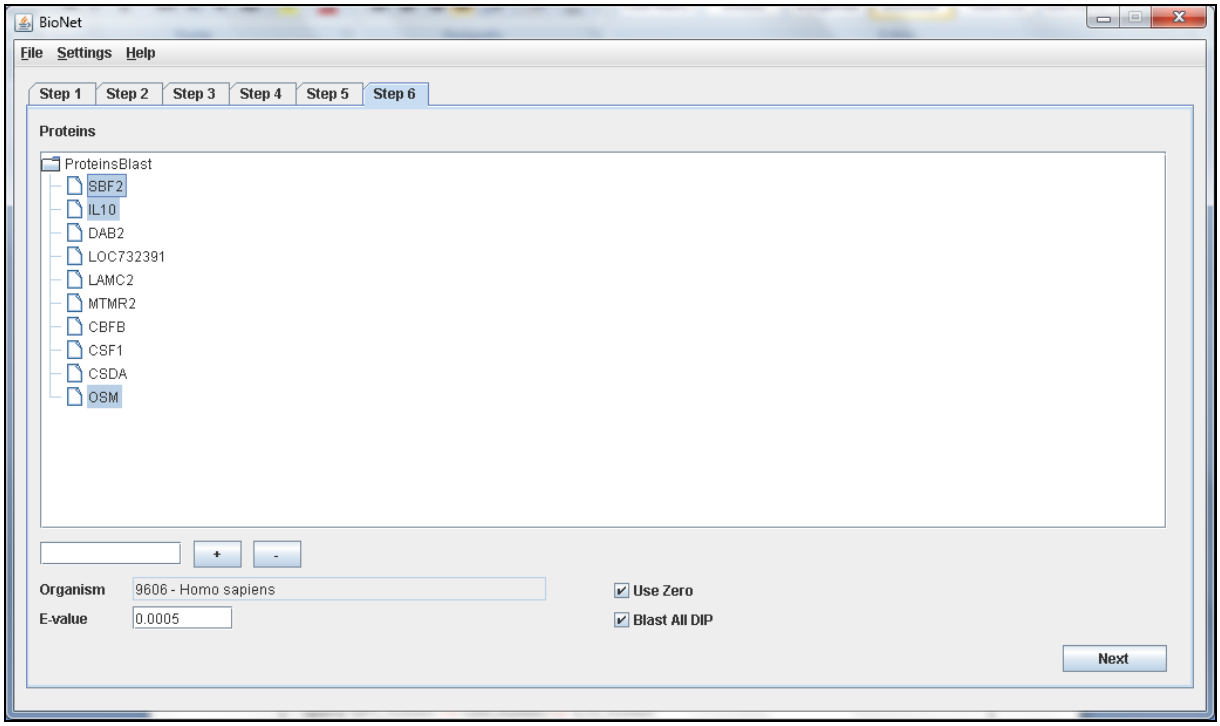

Figura 88 – Seleção das proteínas para pesquisa no PathBLAST

A consulta ao site do PathBLAST foi realizada conforme apresentado na Figura 89 e o resultado do alinhamento das proteínas selecionadas foi apresentado conforme mostra a Figura 90.

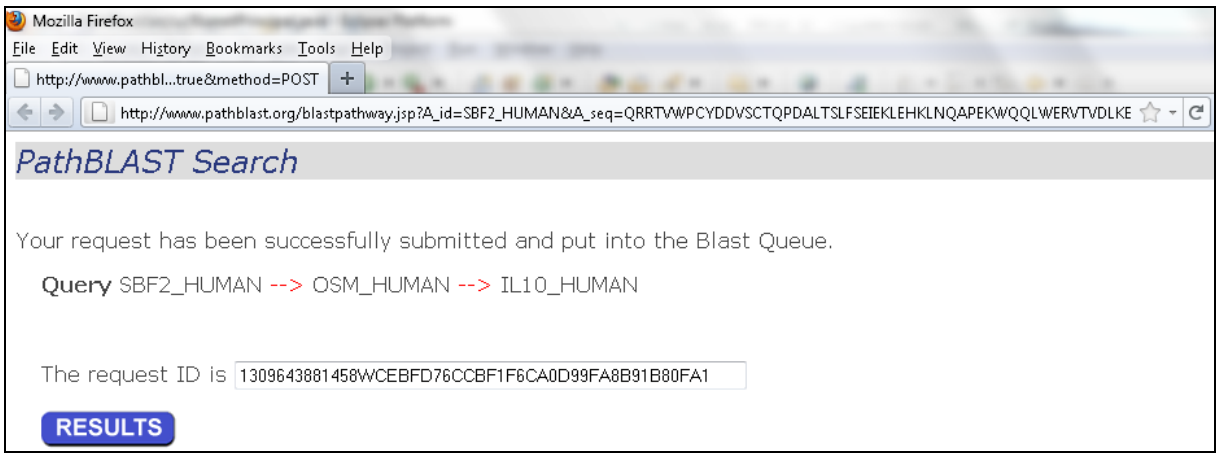

Figura 89 – Consulta das proteínas ao site do PathBLAST

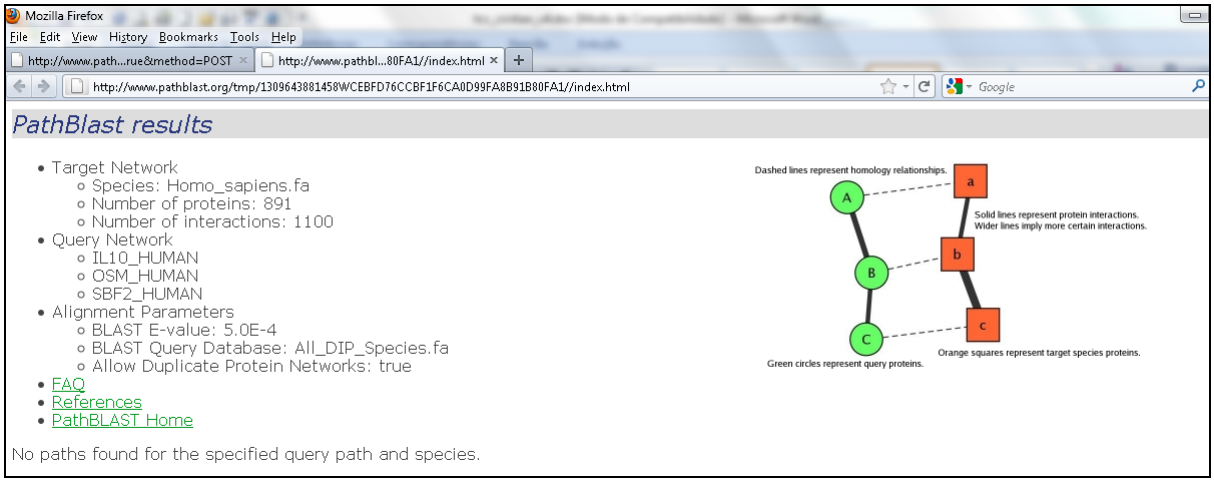

Figura 90 – Resultado da consulta ao site do PathBLAST

5.4 Quarto cenário de testes

No quarto cenário, iniciou-se a pesquisa procurando pelo termo “cockayne”, obtendo assim a lista de ocorrências de doenças com esse termo. Selecionou-se a doença com o identificador “#216400” e com descrição principal “COCKAYNE SYNDROME, TYPE A; CSA” conforme mostra a Figura 91.

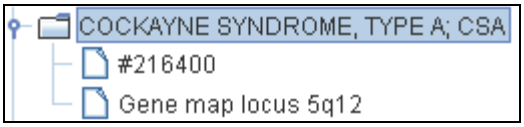

Figura 91 – Doença selecionada no quarto cenário de testes

Após isso foram apresentadas algumas sugestões de proteínas encontradas para a doença selecionada conforme mostra a Figura 92.

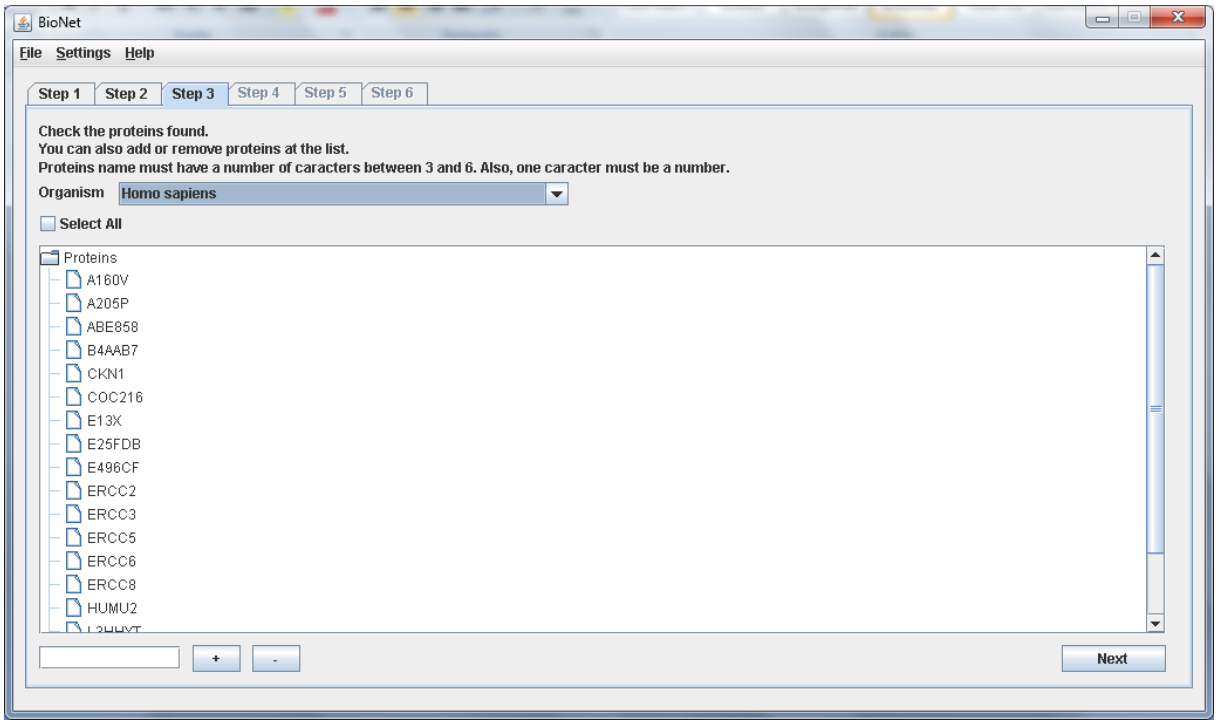

Figura 92 – Lista de sugestões de proteínas que serão pesquisadas

Selecionou-se o organismo “Homo sapiens”. Foram removidos os termos que não eram proteínas e realizou a pesquisa pelas proteínas “CSA”, “CKN1”, “ERCC8”, “CSB”, “ERCC6”, “ERCC3”, “ERCC2”, “ERCC5” e “COFS” conforme mostra a Figura 93.

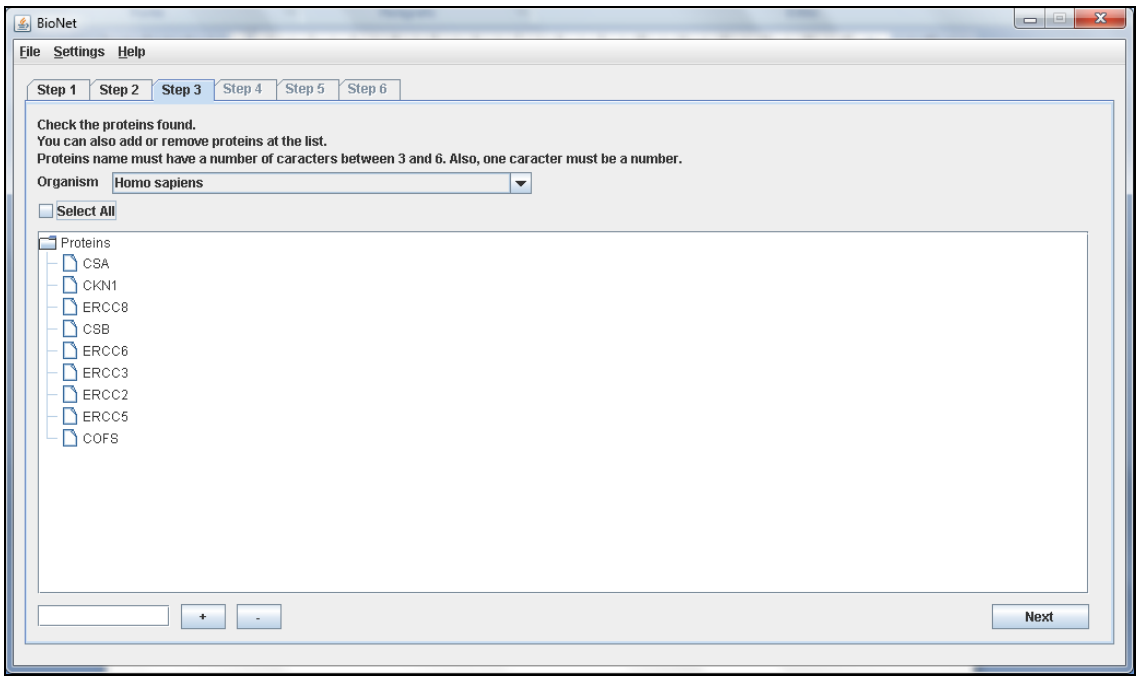

Figura 93 – Lista de proteínas que serão pesquisadas

Obteve-se assim a lista de ocorrências das proteínas em humanos, conforme mostra a Figura 94.

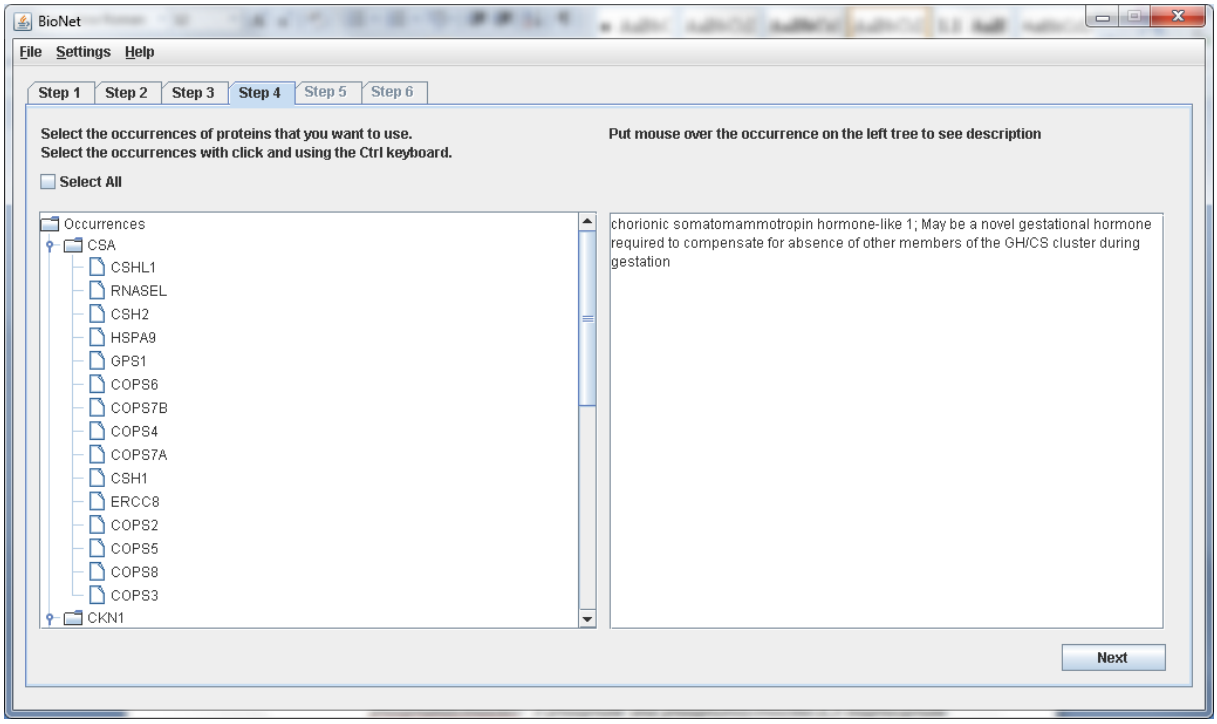

Figura 94 – Lista de ocorrências que serão pesquisadas

Então selecionou-se treze ocorrências de proteínas, cinco da "CSA", uma da "CKN1", uma da "ERCC8", uma da "CSB", uma da "ERCC6", uma da "ERCC3", uma da "ERCC2", uma da "ERCC5" e uma da "COFS", e clicou-se para prosseguir, podendo assim visualizar a rede de interação da proteína, conforme mostra a Figura 95. Abaixo as descrições das proteínas selecionadas, sendo que as repetidas não serão citadas mais de uma vez:

- *HSPA9 heat shock 70kDa protein 9 (mortalin); Implicated in the control of cell proliferation and cellular aging. May also act as a chaperone;*

- *ERCC8 excision repair cross-complementing rodent repair deficiency, complementation group 8; Involved in transcription;*

- *CSH1 chorionic somatomammotropin hormone-like 1; May be a novel gestational hormone required to compensate for absence of other members of the GH/CS cluster during gestation;*

- *COPS7A COP9 constitutive photomorphogenic homolog subunit 4 (Arabidopsis); Component of the COP9 signalosome complex (CSN), a complex involved in various cellular and developmental processes. The CSN complex is an essential regulator of the ubiquitin (Ubl) conjugation pathway by mediating the deneddylation of the cullin subunits of SCF-type E3 ligase complexes, leading to decrease the Ubl ligase activity of SCF-type complexes such as SCF, CSA or DDB2. The complex is also involved in phosphorylation of p53/TP53, c-jun/JUN, IkappaBalpha/NFKBIA, ITPK1 and IRF8/ICSBP, possibly via its association [...];*

- *GPS1 G protein pathway suppressor 1; Essential component of the COP9 signalosome complex (CSN), a complex involved in various cellular and developmental processes. The CSN complex is an essential regulator of the ubiquitin (Ubl) conjugation pathway by mediating the deneddylation of the cullin subunits of SCF-type E3 ligase complexes, leading to decrease the Ubl ligase activity of SCF-type complexes such as SCF, CSA or DDB2. The complex is also involved in phosphorylation of p53/TP53, c-jun/JUN, IkappaBalpha/NFKBIA, ITPK1 and IRF8/ICSBP, possibly via its association with CK2 and PKD kinases. [...];*

- *COPS3 COP9 constitutive photomorphogenic homolog subunit 3 (Arabidopsis); Component of the COP9 signalosome complex (CSN), a complex involved in various cellular and developmental processes. The CSN complex is an essential regulator of the ubiquitin (Ubl) conjugation pathway by mediating the deneddylation of the cullin subunits of SCF-type E3 ligase complexes, leading to decrease the Ubl ligase activity of SCF-type complexes such as SCF, CSA or DDB2. The complex is also involved in phosphorylation of p53/TP53, c-jun/JUN, IkappaBalpha/NFKBIA, ITPK1 and IRF8/ICSBP, possibly via its association [...];*

- *RNASEL ribonuclease L (2',5'-oligoisoadenylate synthetase-dependent); Endoribonuclease, mediator of interferon action, which play a role in mediating resistance to virus infection and apoptosis. Might play a central role in the regulation of mRNA turnover;*

- *CSH2 chorionic somatomammotropin hormone 2; Similar to that of somatotropin;*

- *PGBD3 piggyBac transposable element derived 3;*

- *ERCC3 excision repair cross-complementing rodent repair deficiency, complementation group 3 (xeroderma pigmentosum group B complementing); ATP-dependent 3'-5' DNA helicase, component of the core- TFIIH basal transcription factor, involved in nucleotide excision repair (NER) of DNA and, when complexed to CAK, in RNA transcription by RNA polymerase II. Acts by opening DNA either around the RNA transcription start site or the DNA damage;*

- *ERCC2 excision repair cross-complementing rodent repair deficiency, complementation group 2; ATP-dependent 5'-3' DNA helicase, component of the core-TFIIH basal transcription factor. Involved in nucleotide excision repair (NER) of DNA by opening DNA around the damage, and in RNA transcription by RNA polymerase II by anchoring the CDK-activating kinase (CAK) complex, composed of CDK7, cyclin H and MAT1, to the core-TFIIH complex. Involved in the regulation of vitamin-D receptor activity. Might also have a role in aging process and could play a causative role in the generation of skin cancers;*

- *ERCC5 excision repair cross-complementing rodent repair deficiency, complementation group 5; Single-stranded structure-specific DNA endonuclease involved in DNA excision repair. Makes the 3'incision in DNA nucleotide excision repair (NER). Acts as a cofactor for a DNA glycosylase that removes oxidized pyrimidines from DNA. May also be involved in transcription-coupled repair of this kind of damage, in transcription by RNA polymerase II, and perhaps in other processes too;*

- *CKN2 excision repair cross-complementing rodent repair deficiency, complementation group 6; Is involved in the preferential repair of active genes. Presumed DNA or RNA unwinding function. Corrects the UV survival and RNA synthesis after UV exposure of Cockayne syndrome complementation group B;*

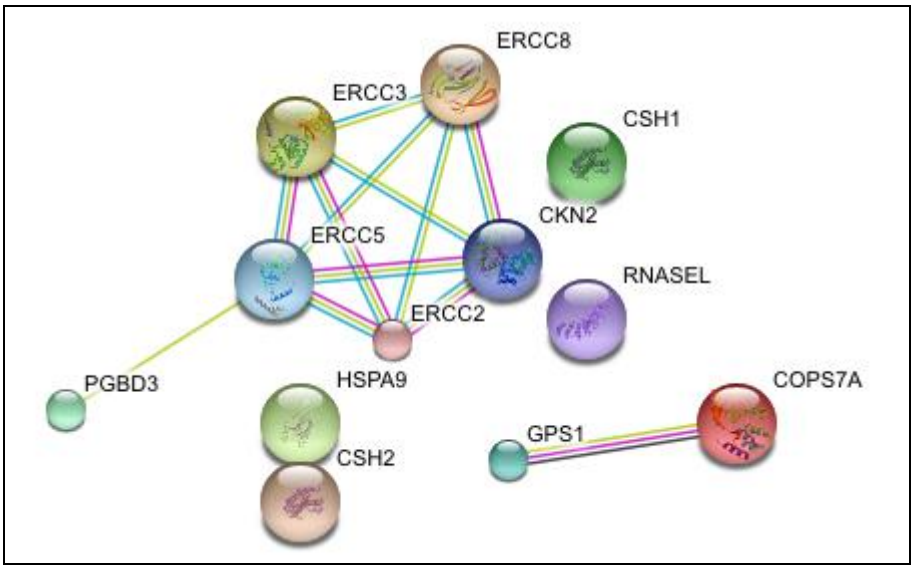

Figura 95 – Quarto cenário de teste rede STRING

Após isso realizou-se o *download* do arquivo XML da rede de interação da proteína, conforme mostra a Figura 96, e importou-se o mesmo no *software* Cytoscape.

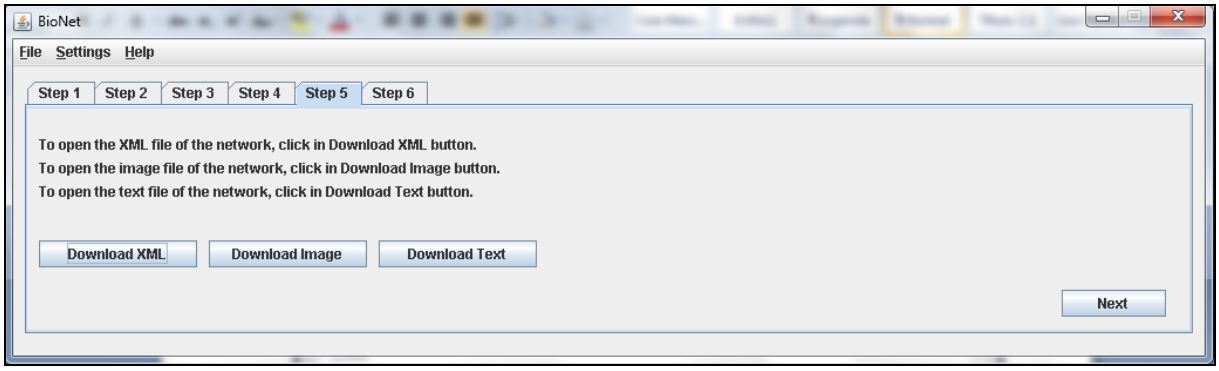

Figura 96 – *Download* do arquivo XML, imagem da rede e arquivo texto

No próximo passo selecionou-se as proteínas “CSH2”, “CSH1” e “CKN2” e clicou-se para prosseguir conforme mostra a Figura 97.

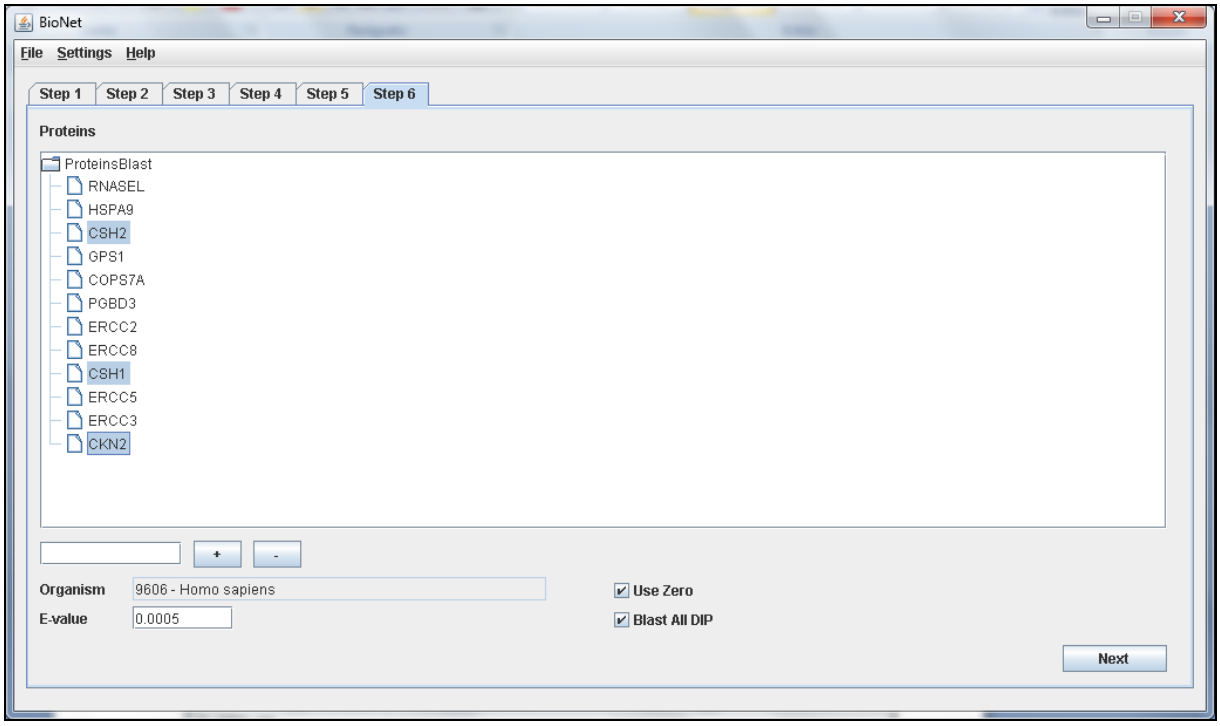

Figura 97 – Seleção das proteínas para pesquisa no PathBLAST

A consulta ao site do PathBLAST foi realizada conforme apresentado na Figura 98, e o resultado do alinhamento das proteínas selecionadas foi apresentado conforme mostra a Figura 99.

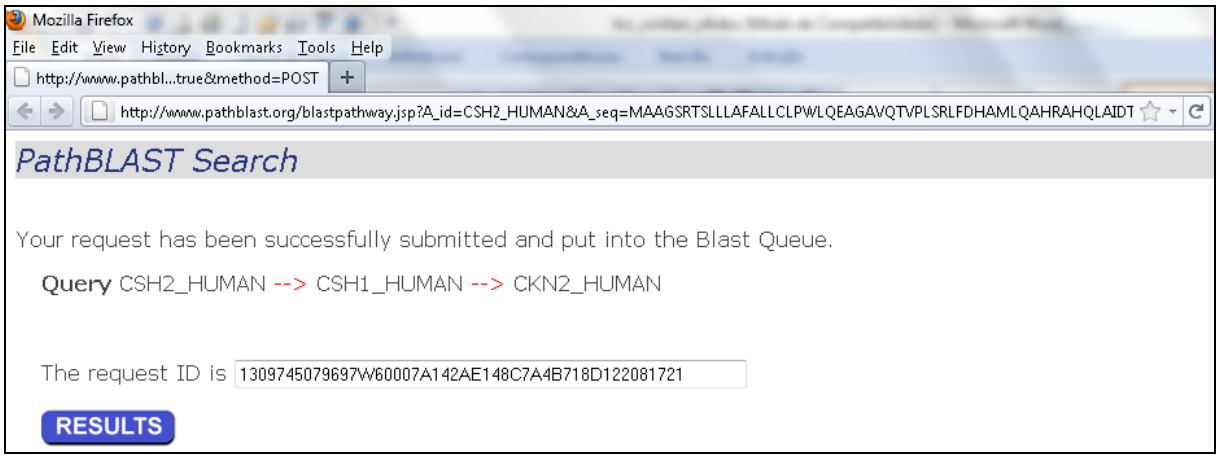

Figura 98 – Consulta das proteínas ao site do PathBLAST

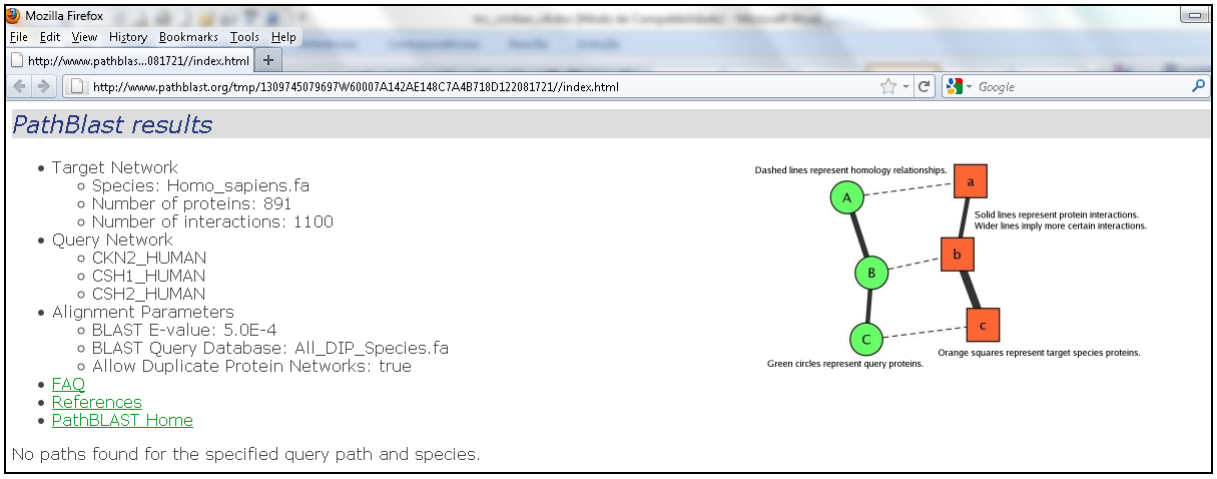

Figura 99 – Resultado da consulta ao site do PathBLAST

5.5 Quinto cenário de testes

No quinto cenário, iniciou-se a pesquisa procurando pelo termo "hepatitis", obtendo assim a lista de ocorrências de doenças com esse termo. Selecionou-se a doença com o identificador "#609532" e com descrição principal "HEPATITIS C VIRUS, SUSCEPTIBILITY TO" conforme mostra a Figura 100.

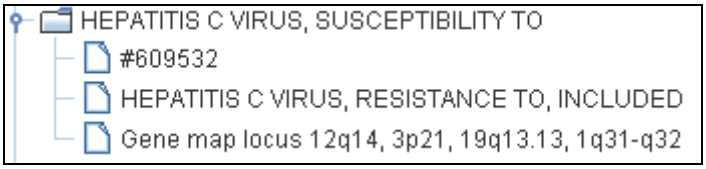

Figura 100 – Doença selecionada no quinto cenário de testes

Após isso foram apresentadas algumas sugestões de proteínas encontradas para a doença selecionada conforme mostra a Figura 101.

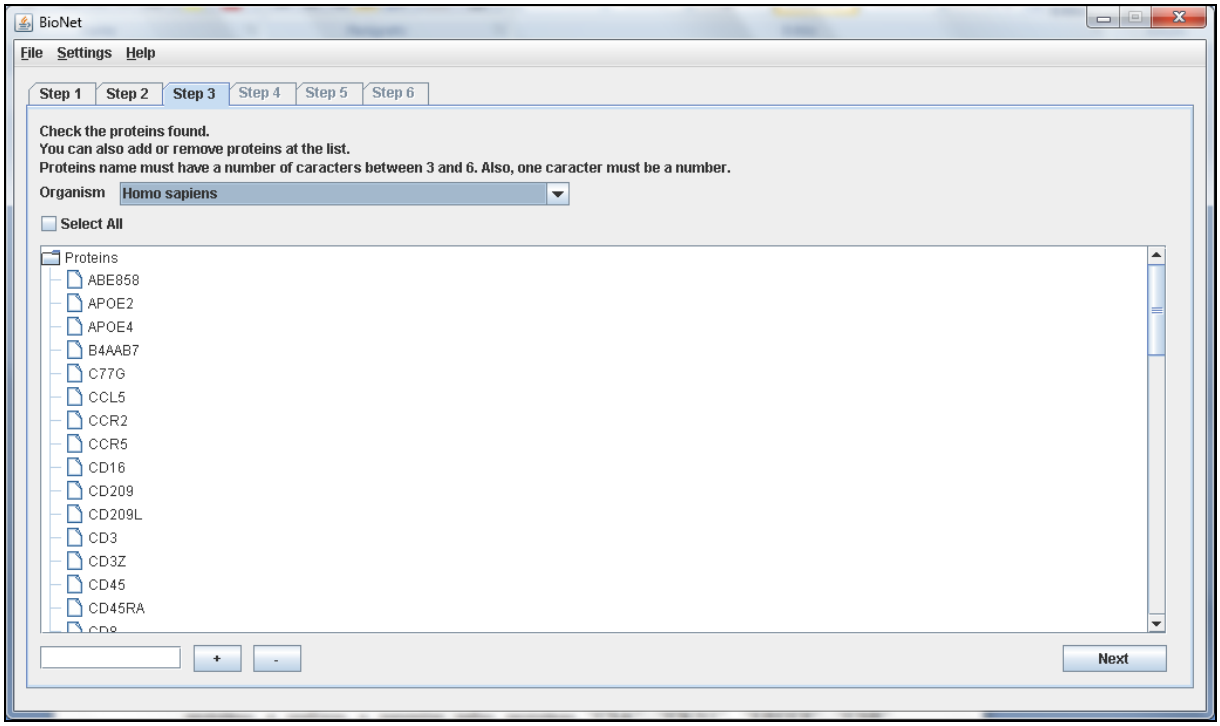

Figura 101 – Lista de sugestões de proteínas que serão pesquisadas

Selecionou-se o organismo “Homo sapiens”, foram removidos os termos que não eram proteínas e realizou-se a pesquisa pelas proteínas “C77G”, “CCL5”, “CCR2”, “CCR5”, “CD16”, “DGAT2”, “EPHA2”, “ERK1” e “IL28B”, conforme mostra a Figura 102.

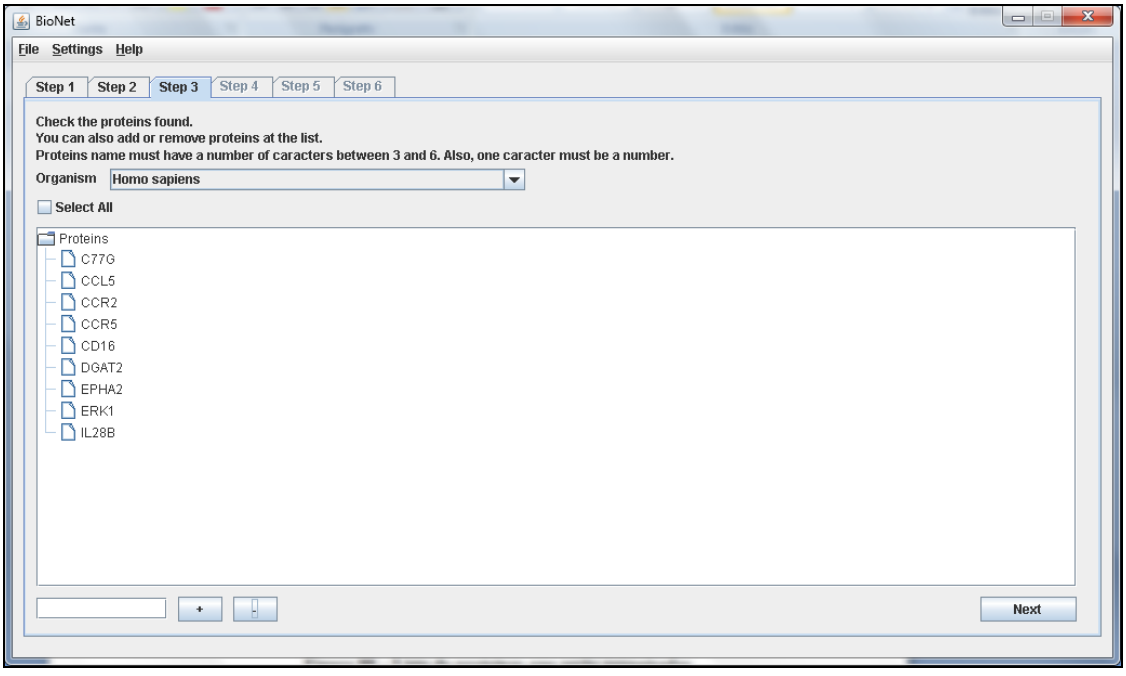

Figura 102 – Lista de proteínas que serão pesquisadas

Obteve-se assim a lista de ocorrências das proteínas em humanos, conforme mostra a Figura 103.

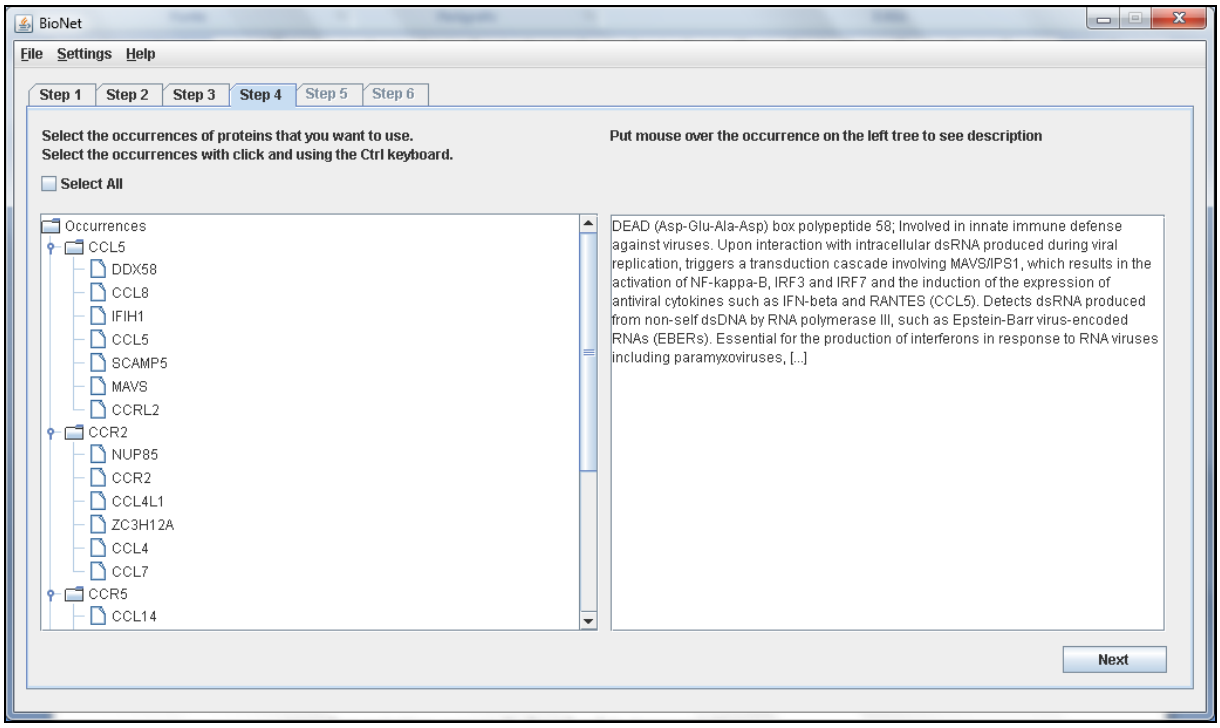

Figura 103 – Lista de ocorrências que serão pesquisadas

Então selecionou-se dez ocorrências de proteínas, três da “CCL5”, uma da “CCR2”, uma da “CD16”, uma da “DGAT2”, uma da “EPHA2”, uma da “ERK1”, duas da “IL28B”, e clicou para prosseguir, podendo-se assim visualizar a rede de interação da proteína, conforme mostra a Figura 104. Abaixo as descrições das proteínas selecionadas, sendo que as repetidas não serão citadas mais de uma vez:

- *DDX58 DEAD (Asp-Glu-Ala-Asp) box polypeptide 58; Involved in innate immune defense against viruses. Upon interaction with intracellular dsRNA produced during viral replication, triggers a transduction cascade involving MAVS/IPS1, which results in the activation of NF-kappa-B, IRF3 and IRF7 and the induction of the expression of antiviral cytokines such as IFN-beta and RANTES (CCL5). Detects dsRNA produced from non-self dsDNA by RNA polymerase III, such as Epstein-Barr virus-encoded*

RNAs (EBERs). Essential for the production of interferons in response to RNA viruses including paramyxoviruses, [...];

- *IFIH1 interferon induced with helicase C domain 1; RNA helicase that, through its ATP-dependent unwinding of RNA, may function to promote message degradation by specific RNases. Seems to have growth suppressive properties. Involved in innate immune defense against viruses. Upon interaction with intracellular dsRNA produced during viral replication, triggers a transduction cascade involving MAVS/IPS1, which results in the activation of NF-kappa-B, IRF3 and IRF7 and the induction of the expression of antiviral cytokines such as IFN-beta and RANTES (CCL5). ATPase activity is specifically induce [...];*

- *MAVS mitochondrial antiviral signaling protein; Required for innate immune defense against viruses. Acts downstream of DDX58 and IFIH1/MDA5, which detect intracellular dsRNA produced during viral replication, to coordinate pathways leading to the activation of NF-kappa-B, IRF3 and IRF7, and to the subsequent induction of antiviral cytokines such as IFN-beta and RANTES (CCL5). May activate the same pathways following detection of extracellular dsRNA by TLR3. May protect cells from apoptosis;*

- *CCL4 chemokine (C-C motif) ligand 4; Monokine with inflammatory and chemokinetic properties. Binds to CCR5. One of the major HIV-suppressive factors produced by CD8+ T-cells. Recombinant MIP-1-beta induces a dose-dependent inhibition of different strains of HIV-1, HIV-2, and simian immunodeficiency virus (SIV). The processed form MIP-1-beta(3-69) retains the abilities to induce down-modulation of surface expression of the chemokine receptor CCR5 and to inhibit the CCR5-mediated entry of HIV-1 in T-cells. MIP-1-beta(3-69) is also a ligand for CCR1 and CCR2 isoform B;*

- *GNG8 guanine nucleotide binding protein (G protein), gamma 8; Guanine nucleotide-binding proteins (G proteins) are involved as a modulator or transducer in various transmembrane signaling systems. The beta and gamma chains are required for the*

GTPase activity, for replacement of GDP by GTP, and for G protein- effector interaction;

- *DGAT1 diacylglycerol O-acyltransferase homolog 1 (mouse); Catalyzes the terminal and only committed step in triacylglycerol synthesis by using diacylglycerol and fatty acyl CoA as substrates. In contrast to DGAT2 it is not essential for survival. May be involved in VLDL (very low density lipoprotein) assembly;*

- *EFNA1 ephrin-A1; Plays an important role in angiogenesis and tumor neovascularization. The recruitement of VAV2, VAV3 and PI3-kinase p85 subunit by phosphorylated EPHA2 is critical for EFNA1-induced RAC1 GTPase activation and vascular endothelial cell migration and assembly (By similarity). Exerts anti-oncogenic effects in tumor cells through activation and down-regulation of EPHA2. Activates EPHA2 by inducing tyrosine phosphorylation which leads to its internalization and degradation. Acts as a negative regulator in the tumorigenesis of gliomas by down-regulating EPHA2 and FAK. Can evoke co [...];*

- *DUSP5 dual specificity phosphatase 5; Displays phosphatase activity toward several substrates. The highest relative activity is toward ERK1;*

- *IL28B interleukin 28B (interferon, lambda 3); Cytokine with immunomodulatory activity. Up-regulates MHC class I antigen expression. Displays potent antiviral activity. Also displays antitumor activity. Ligand for the heterodimeric class II cytokine receptor composed of IL10RB and IL28RA. The ligand/receptor complex seems to signal through the Jak-STAT pathway;*

- *IL28A interleukin 28A (interferon, lambda 2); Cytokine with immunomodulatory activity. Up-regulates MHC class I antigen expression. Displays potent antiviral activity. Also displays antitumor activity. Ligand for the heterodimeric class II cytokine receptor composed of IL10RB and IL28RA. The ligand/receptor complex seems to signal through the Jak-STAT pathway;*

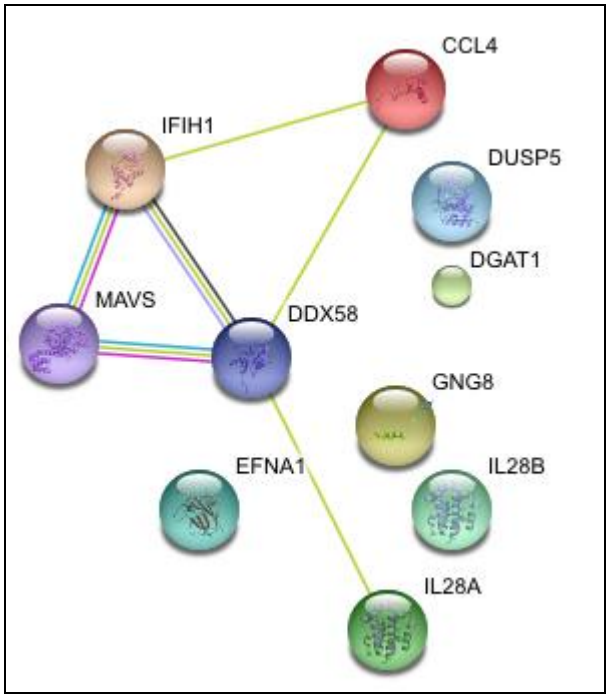

Figura 104 – Quinto cenário de testes rede STRING

Após isso realizou-se o *download* do arquivo XML da rede de interação da proteína, conforme mostra a Figura 105, e importou-se o mesmo no *software* Cytoscape.

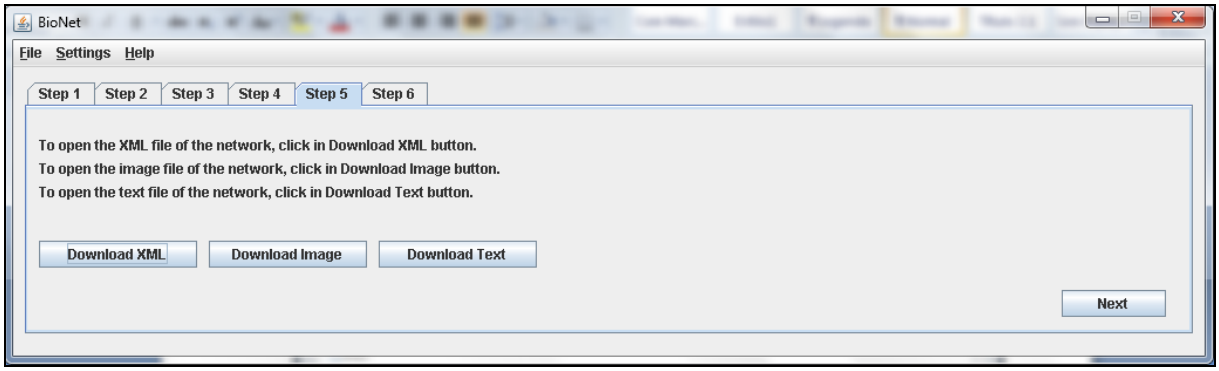

Figura 105 – *Download* do arquivo XML, imagem da rede e arquivo texto

No próximo passo selecionou-se as proteínas “CCL4”, “IL28A” e “IL28B” e clicou-se para prosseguir, conforme mostra a Figura 106.

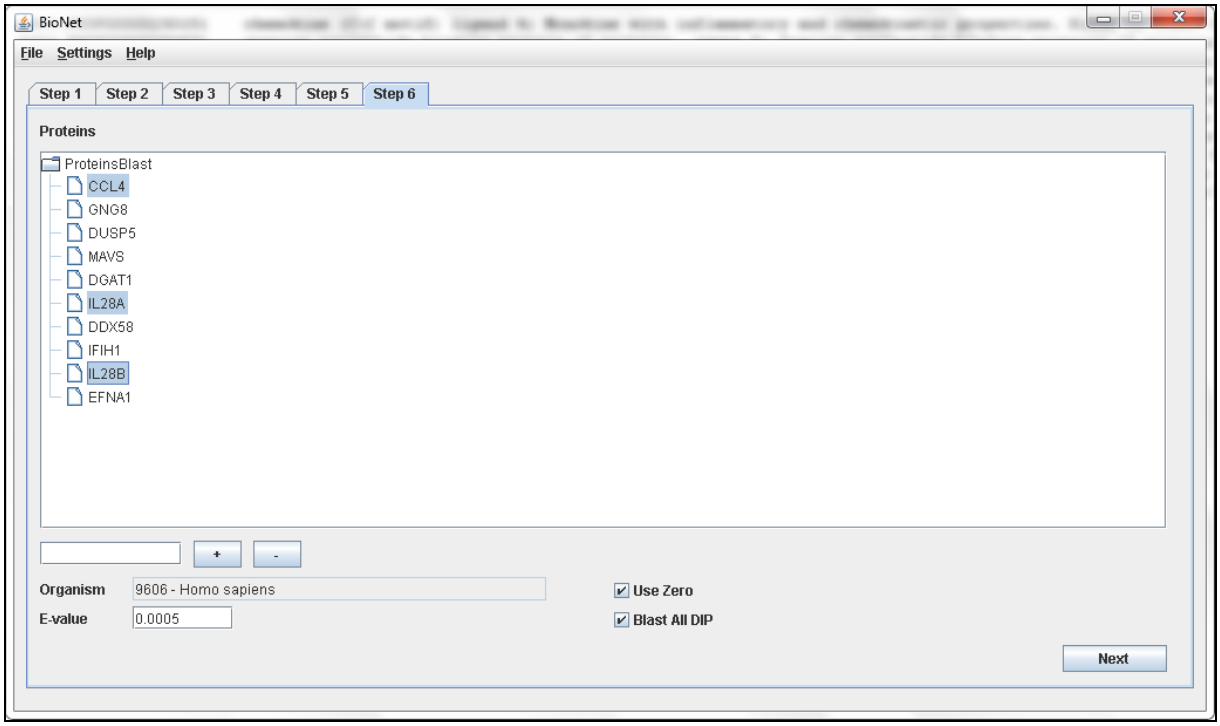

Figura 106 – Seleção das proteínas para pesquisa no PathBLAST

A consulta ao site do PathBLAST foi realizada apresentando-se a Figura 107, e o resultado do alinhamento das proteínas selecionadas foi apresentado conforme mostra a Figura 108.

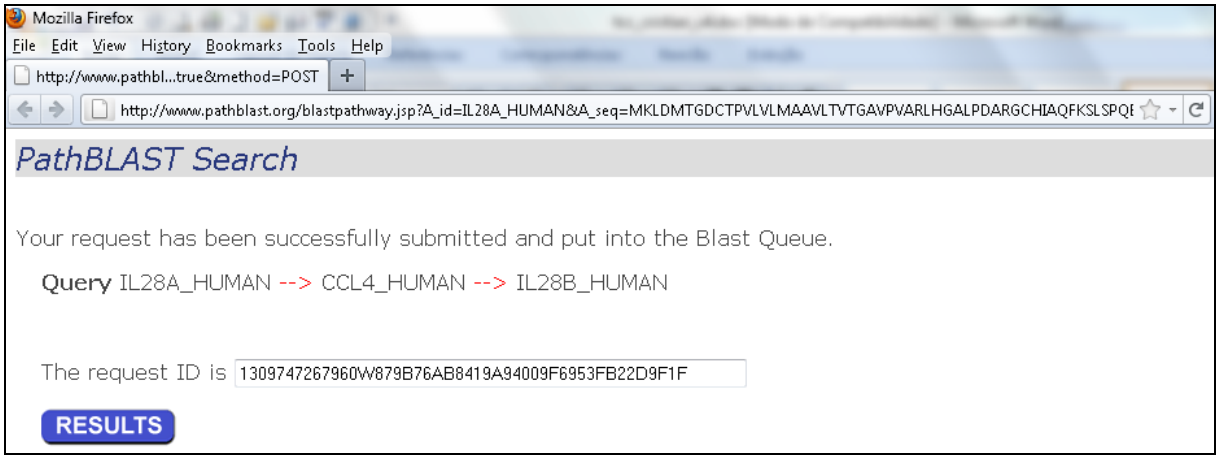

Figura 107 – Consulta das proteínas ao site do PathBLAST

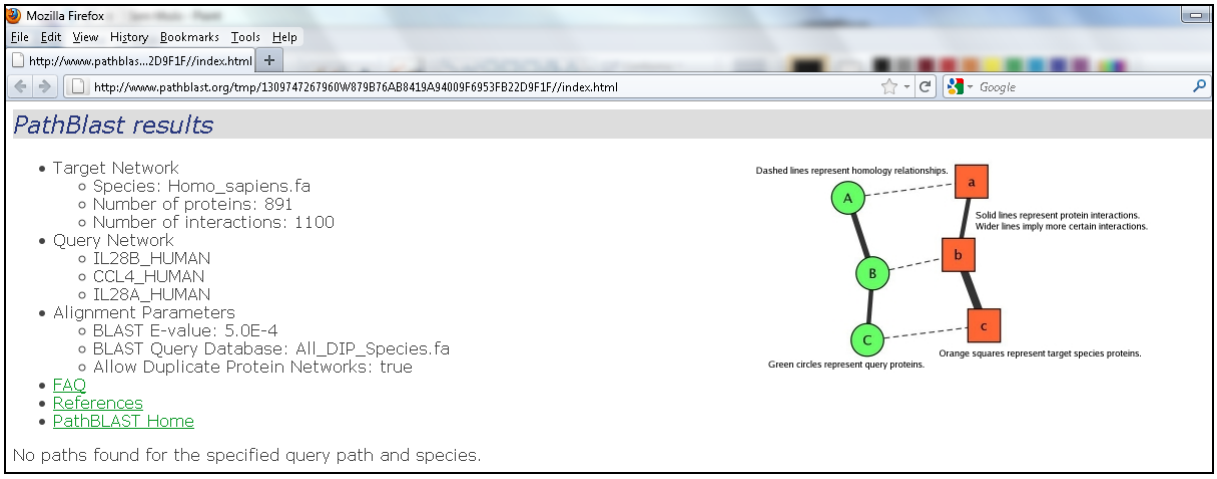

Figura 108 – Resultado da consulta ao site do PathBLAST

5.6 Considerações finais

Os cenários de testes foram os mesmos realizados no trabalho prévio (Oldra, 2009) e apresentaram resultados próximos ou praticamente iguais aos testes. Mesmo com a aplicação de expressões regulares o sistema apresenta diversas sugestões de proteínas que não são proteínas. Esse problema pode ser contornado utilizando o cadastro de proteínas que devem ser desconsideradas conforme mostra a Figura 109.

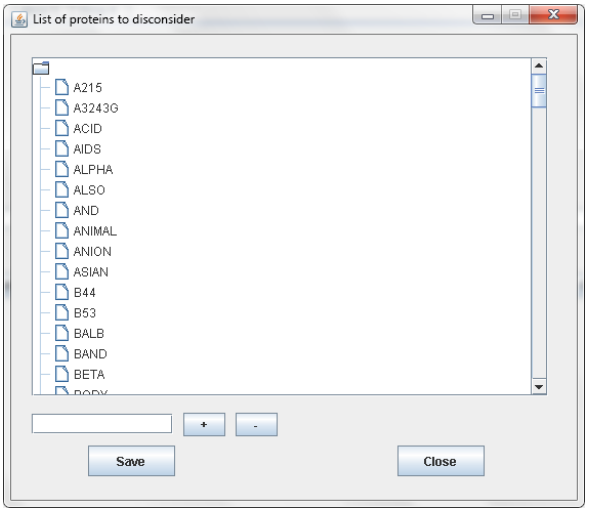

Figura 109 – Cadastro de proteínas a serem desconsideradas

6 CONCLUSÃO

Neste trabalho desenvolveu-se uma ferramenta de *software* que permite ao usuário da área da biologia e bioinformática consultar e visualizar informações biológicas sem a necessidade do acesso individual aos sites dos bancos de dados biológicos OMIM, STRING centralizando assim o fluxo de pesquisa das doenças.

A nova funcionalidade possibilitou a integração ao *site* do PathBLAST para consulta ao alinhamento de proteínas.

O sistema recebeu melhorias de usabilidade, e permite agora navegar entre as abas da aplicação simulando o workflow apresentado. A seguir serão apresentadas mais algumas melhorias que foram implementadas neste trabalho:

- Consulta ao *log* da aplicação onde o usuário poderá verificar os métodos, urls e chamadas utilizadas pelo sistema;

- Cadastro das proteínas que devem ser desconsideradas;

- Cadastro do *browser* que será utilizado para abertura de urls, o caminho do diretório temporário da aplicação, o caminho do arquivo de *log*, o caminho do arquivo que contém a lista de proteínas que serão desconsideradas, o caminho do arquivo que contém a lista de organismos que serão carregados.

Para trabalhos futuro seria interessante analisar as respostas dos especialistas com objetivo de aperfeiçoar o sistema e recomendar outras formas de consulta e manipulação dos dados.

REFERÊNCIAS

Algo Sobre Vestibular e Concursos. Disponível em: <http://www.algosobre.com.br/biologia/dna-e-rna.html>. Acesso em maio de 2011.

Brasil Escola - Bioinformática. Disponível em: <http://www.brasilescola.com/biologia/bioinformatica.htm>. Acesso em março de 2011.

BEBEK, G.; YANG, J. **Pathfinder: mining signal transduction pathway segments from protein-protein interaction networks**. BMC Bioinformatics, doi:10.1186/1471-2105-8-335, v.8, n.335, 2007.

Biologia Molecular. Disponível em: <http://www.biomol.org/>. Acesso em fevereiro de 2011.

BRITO, Rogério Theodoro De. **Alinhamento de Sequências Biológicas**. Dissertação (Mestrado Ciência da Computação). Universidade de São Paulo. São Paulo, 2003.

Human Genome Project. **Bioinformatics: Human Genome Research in Progress**. Disponível em: <http://www.ornl.gov/sci/techresources/Human_Genome/research/informatics.shtml>. Acesso em: março de 2011.

LEMOS, Melissa. **Workflow para Bioinformática.** Tese (Doutor em Informática). Pontifica Universidade Católica (PUC). Rio de Janeiro, 2004.

MedicineNet. **Genetic Diseases Overview**. Disponível em: <http://www.medicinenet.com/genetic_disease/article.htm>. Acesso em: março de 2011.

Oldra, Samuel. **Integração de Dados Biológicos – Proteínas e Doenças Gênicas**. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade de Caxias do Sul.

Passarge, Eberhard. **Genética: texto e atlas** – 2. ed. Artmed Editora, 20-45 (2004).

Prosdocimi, F. et al. **Bioinformática: manual do usuário**. Biotecnologia, Ciência & Desenvolvimento. Ano 5. V 29. nov/dez 2002. p. 12-25. Disponível em: <http://www.biotecnologia.com.br/revista/bio29/bioinfo.pdf>. Acesso em fevereiro de 2011.

Raychaudhuri, Soumya. **Computacional Text Analysis for Functional Genomics and Bionformatics**. Oxford University Press, 42-52 (2006).

ROCHA, Cícero Pinho. **Bancos de Dados em Bioinformática.** Licenciatura plena em Ciências da Computação. Piauí, 2007.

SILVA, Fabrício N. Da. **In Services: Um sistema para gerenciamento de dados intermediários em workflows científicos na bioinformática.** Dissertação (Mestrado em Sistemas e Computação). Instituto Militar de Engenharia. Rio de Janeiro, 2006.

WfMC – Workflow Management Coalition, The Workflow Handbook 2004, Fischer,L.(ed.). Disponível em: <http://www.wfmc.org/information/handbook04.htm>.