Edson Luiz Folador	GO-SIEVe: Software para determinar códigos de evidência em anotação gênica	2008

EDSON LUIZ FOLADOR

GO-SIEVe: SOFTWARE PARA DETERMINAR CÓDIGOS DE EVIDÊNCIA EM ANOTAÇÃO GÊNICA

Dissertação apresentada ao Programa de Pós-graduação em Tecnologia de Saúde da Pontifícia Universidade Católica do Paraná, na área de Bioinformática, linha de pesquisa de Sistemas de Informação, como pré-requisito parcial para a obtenção do título de Mestre.

CURITIBA

2008

EDSON LUIZ FOLADOR

GO-SIEVe: SOFTWARE PARA DETERMINAR CÓDIGOS DE EVIDÊNCIA EM ANOTAÇÃO GÊNICA

Dissertação apresentada ao Programa de Pós-graduação em Tecnologia de Saúde da Pontifícia Universidade Católica do Paraná, na área de Bioinformática, linha de pesquisa de Sistemas de Informação, como pré-requisito parcial para a obtenção do título de Mestre.

CURITIBA

2008

FOLHA DE APROVAÇÃO

Edson Luiz Folador

GO-SIEVe: Software para determinar códigos de evidência em anotação gênica

Dissertação apresentada ao Programa de Pós-graduação em Tecnologia de Saúde da Pontifícia Universidade Católica do Paraná, na área de Bioinformática, linha de pesquisa de Sistemas de Informação, como pré-requisito parcial para a obtenção do título de Mestre.

Aprovado em 19 de dezembro de 2008.

Banca examinadora

Prof. Dr. Humberto Maciel França Madeira
Pontifícia Universidade Católica do Paraná
Julgamento: _____

Assinatura: _____

Prof. Dra. Andreia Malucelli
Pontifícia Universidade Católica do Paraná
Julgamento: _____

Assinatura: _____

Prof. Dr. Gerson Linck Bichinho
Pontifícia Universidade Católica do Paraná
Julgamento: _____

Assinatura: _____

Prof. Dr. Leonardo Magalhães Cruz
Universidade Federal do Paraná
Julgamento: _____

Assinatura: _____

Dedico este trabalho aos meus pais que, sem sequer terem concluído o ensino primário (hoje ensino fundamental), sempre me incentivaram a estudar e, no caso deste mestrado, deram ajuda financeira, sem a qual não seria possível realizar este trabalho. Dedico também aos meus filhos Jiuliane e Eduardo por entenderem a minha ausência quando me dedicando aos estudos.

AGRADECIMENTOS

À Deus, acima de tudo, pela oportunidade e condições dadas.

Ao meu orientador pela dedicação, paciência e conhecimentos biológicos ensinados.

A minha co-orientadora pelo incentivo, pela motivação nas horas difíceis e pelos conhecimentos computacionais ensinados.

A ambos orientadores pela maneira como organizaram e conduziram este trabalho, pelo bom humor e carisma em todos os momentos de orientação e pelo auxílio em todas as etapas deste trabalho.

A todos os demais professores do programa de mestrado que, direta ou indiretamente, seja com sugestões, opiniões, críticas, aulas, materiais de pesquisa ou com uma palavra amiga contribuíram para a realização deste trabalho.

Aos especialistas que validaram este trabalho, Dr. Humberto Maciel França Madeira e Dr. Leonardo Magalhães Cruz, pelo tempo e esforço despendido à leitura de 1.884 resumos de artigos.

Aos meus colegas de estudo pela amizade, pelo carinho, pelo momento de descontração, pela troca de informações, pela ajuda nos laboratórios, pelas dicas, pela torcida e principalmente pela força dada no sentido de concluir este trabalho.

Como a ciência moderna cresce em complexidade e alcance, a necessidade de maior colaboração entre cientistas de instituições diferentes, em diferentes áreas, e em todas as disciplinas científicas torna-se cada vez mais importante.

James Hendler

RESUMO

Durante o processo de anotação gênica, os especialistas precisam verificar um grande volume de informações para que, junto com seus conhecimentos sobre o organismo, atribuam uma função a um gene. Como essas informações são utilizadas no processo de anotação em outros projetos genoma, é necessário que sejam consistentes, a fim de que possíveis inconsistências não sejam replicadas para outros projetos. Mesmo existindo qualificadores nos projetos genomas que informam o estado de uma anotação, eles podem não revelar como o processo de anotação foi executado ou, o que o validador considerou ao analisar uma anotação automática e atribuir uma função ao gene. Procurar, porém, neste grande volume de dados, por informações que possam fornecer alguma evidência para caracterizar um gene não é uma tarefa simples para um curador, exigindo também recursos informatizados. Para executar esta tarefa, foi desenvolvido o software GO-SIEVe, para (1) armazenar a literatura acessível pelo BLAST para um gene; (2) procurar por termos predefinidos e relacionados aos códigos de evidência definidos pelo GO (The Gene Ontology, 2008) que possam revelar o provável fundamento utilizado pelo anotador ao atribuir função ao gene e, (3) atribuir o código de evidência correspondente ao termo encontrado junto às informações do gene. Para validar o GO-SIEVe, foi usado o genoma da bactéria *Chromobacterium violaceum*, anotado na plataforma SABIÁ. Para os 54 termos cadastrados e os 4.431 genes de *C. violaceum* processados pelo GO-SIEVe, foram baixadas e armazenadas na base de dados 181.098 literaturas, sendo que destas, 105.204 (58,1%) acumularam a atribuição de 121.115 códigos de evidência (CV=, IDA, IEP, IGC, IGI, IMP e ISS), correspondendo a 39 dos termos cadastrados no GO-SIEVe. Destas 121.115 literaturas, somente 2.517 não eram repetidas (únicas) e, deste montante, cada especialista validou o total de 1.889 literaturas (75,0%), as quais correspondem a média de 82,1% de acerto pelo GO-SIEVe, mostrando-se eficiente ao atribuir os códigos de evidência aos genes de *C. violaceum*. O GO-SIEVe pode ser um importante módulo complementar às plataformas de anotação existentes, pois pode filtrar a literatura existente de um

gene e minimizar a quantidade de dados que um curador precisaria ler para atribuir uma função ao gene e, por conseqüência, reduzir tempo e custo envolvido na anotação ou re-anotação de um genoma. Além da inserção dos códigos de evidência em um genoma, o GO-SIEVe disponibiliza o(s) termo(s) e a(s) literatura(s) relacionadas junto às informações do gene, fornecendo ao usuário da anotação deste gene, informações que dão suporte à anotação.

Palavras-chave: Bioinformática, Anotação Gênica, Código de Evidência, Anotação Automática, Chromobacterium violaceum.

ABSTRACT

During the process of gene annotation, specialists must verify a large amount of information so that, together with their knowledge about the organism, assign a function to a gene. As these information are used in the process of annotation on other genome projects, they must be consistent, so that possible inconsistencies can not be replicated in other projects. Even having qualifiers at genomes projects that inform the state of an annotation, they can not prove how the annotation process was executed, or what the valuator considered when examining an automatic annotation and assigned a function to the gene. Search, however, in this large volume of data, for information that may provide some evidence to identify a gene is not a simple task for a curator, which also requires computer resources. To accomplish this task, the software GO-SIEVe was developed, (1) to store the literature acessible by BLAST for a gene, (2) to search for pre-defined terms and connected with the evidence code established by the GO (The Gene Ontology, 2008) that may reveal the probable plea used by anotador to assign the gene function, and (3) to assign the corresponding evidence code to the term found at the gene information. To validate the GO-SIEVe, was used the genome of the bacterium Chromobacterium violaceum, annotaded in SABIÁ platforms. For all 54 terms registered and 4,431 genes of C. violaceum processed by GO-SIEVe, there were downloaded and stored in the database 181,098 literature, and from those, 105,204 (58,1%) accumulated the assignment of 121,115 codes of evidence (CV=, IDA, IEP, IGC, IGI, IMP e ISS), corresponding to 39 of the terms registered in GO-SIEVe. From 121,115 literature, only 2,517 were not repeated (unique). From this amount, each experts validated the total of 1,889 literature (75.0%), which showed an average of 82.1% hit by GO-SIEVe, showing efficiency to assign the evidence codes at genes of C. violaceum. GO-SIEVe module may be an important complement to the existing annotation platforms, since it can filter the existing literature of a gene and minimize the amount of data that a curator would have read to assign a function to the gene and, consequently, reduce time and cost involved in the annotation or re-annotation of a genome. Besides the inclusion of

evidence codes in a genome, the GO-SIEVe makes available the terms and literature connected together with the gene information, providing the user of the annotation of this gene, information that give support to the annotation.

Key-words: Bioinformatic, Genome Annotation, Evidence Code, Automatic Annotation, Chromobacterium violaceum.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE GRÁFICOS

LISTA DE SIGLAS E ABREVIAÇÕES

A	Base nitrogenada – adenina
BBP	*Brucella Bioinformatics Portal*
BD	Base de Dados
BLAST	*Basic Local Alignment Search Tool*
BP	Processo Biológico, categoria do Gene Ontology
BRGENE	Rede Nacional de Seqüenciamento
C	Base nitrogenada – citosina
CAM	*Contig Analysis Manager*
CC	Componente Celular, categoria do Gene Ontology
CNPq	Conselho Nacional de Desenvolvimento Científico e Tecnológico
Cv	*Chromobacterium violaceum*
DNA	Ácido desóxi-ribonucléico (ADN)
EMBL	*European Molecular Biology Laboratory*
EXP	*Inferred from Experiment,* código de evidência definido no GO
FANTOM2	*Functional Annotation of Mouse*
FTP	*File Transfer Protocol*
G	Base nitrogenada – guanina
GAIA	Software para *Genome Annotation and Information Analysis*
Genopar	Projeto Genoma do Paraná
GO	*Gene Ontology*
GO-SIEVe	GO-*based Software for Inferring Evidence Codes of Annotated Genes*
HGP	*Human Genome Project*
IA	Inteligência artifical
IC	*Inferred by Curator,* código de evidência definido no GO
IDA	*Inferred from Direct Assay,* código de evidência definido no GO
IEA	*Inferred from Electronic Annotation,* código de evidência definido no GO
IEP	*Inferred from Expression Pattern,* código de evidência definido no GO
IGC	*Inferred from Genomic Context,* código de evidência definido no GO
IGI	*Inferred from Genetic Interaction,* código de evidência definido no GO
IMP	*Inferred from Mutant Phenotype,* código de evidência definido no GO
IPI	*Inferred from Physical Interaction,* código de evidência definido no GO
ISA	*Inferred from Sequence Alignment,* código de evidência definido no GO

ISM	*Inferred from Sequence Model,* código de evidência definido no GO
ISO	*Inferred from Sequence Orthology,* código de evidência definido no GO
ISS	*Inferred from Sequence or Structural Similarity,* código de evidência definido no GO
html	*hyper text markup language*
http	*hyper text transfer protocol*
JDBC	*Java Data Base Connectivity*
JUDE	Java and *UML Developers' Environment*
LNCC	Laboratório Nacional de Computação Científica
MF	Função Molecular, categoria do Gene Ontology
NAS	*Non-traceable Author Statement,* código de evidência definido no GO
NCBI	*National Center for Biotechnology Information*
ND	*No biological Data available,* código de evidência definido no GO
NR	*Not Recorded,* código de evidência definido no GO
ONSA	*Organization for Nucleotide Sequencing and Analysis*
ORF	*Open Reading Frames*
PAM	*Protein Analysis Manager*
Pb	Pares de bases
PERL	*Practical Extraction and Report Language*
PGH	Projeto Genoma Humano
PIGS	Projeto Genoma Sul
PLN	Processamento de Linguagem Natural
PUCPR	Pontifícia Universidade Católica do Paraná
RCA	*Inferred from Reviewed Computational Analysis,* código de evidência definido no GO
SABIÁ	*System for Automated Bacterial Integrated Annotation*
SGBD	Sistema de Gerenciamento de Banco de Dados
SQL	*Structured Query Language* – Linguagem de consulta estruturada
T	Base nitrogenada – timina
TAS	*Traceable Author Statement,* código de evidência definido no GO
U	Base nitrogenada – uracila
UML	*Unified Model Language*
Unicamp	Universidade Estadual de Campinas

SUMÁRIO

1 INTRODUÇÃO

Em projetos de seqüenciamento de genomas, após a obtenção das seqüências e a localização dos potenciais genes, procura-se atribuir uma função a cada um dos genes, de forma a permitir melhor entendimento da biologia dos organismos sob estudo. Este processo de atribuição de função aos genes é parte de um processo denominado anotação funcional ou anotação gênica que é feito com uso da bioinformática, apoiada ou não por processos experimentais.

A determinação experimental da função dos genes depende da aplicação de diversos procedimentos laboratoriais, muitas vezes de custo elevado e de alta demanda de tempo de trabalho. Já os processos que fazem uso de bioinformática são mais rápidos e uma grande quantidade de genes pode ser anotada automaticamente usando recursos computacionais. A lógica deste processo está fundamentada na similaridade existente entre os genes. Se dois genes são similares, ou têm a seqüência de aminoácidos e domínios conservados com relação à de outro gene, deduz-se que ambos possuem funções equivalentes. Computacionalmente o processo é executado a partir da seqüência de um gene de interesse (Gi). Compara-se este gene (Gi) com vários outros genes (Gb) depositados em bases de dados públicas mundialmente conhecidas pelos anotadores e, se for encontrado um gene similar (Gb) sua função é atribuída ao gene pesquisado (Gi), dizendo que ambos têm a mesma função.

A anotação de um genoma usando busca por similaridade é executada em duas etapas: uma automática e outra manual, onde a confiabilidade desta anotação dependerá de dois fatores: (1) de quão similares sejam as seqüências entre o gene comparado e os genes encontrados (Gi e Gb) e, (2) de quão confiável foi a anotação dos genes depositados nestas bases de dados públicas (Gb). Se a anotação do gene encontrado nas bases de dados públicas (Gb) foi baseada em experimentação, isto dá mais confiabilidade à anotação, mas se for dependente apenas de similaridade em nível de seqüência, essa confiabilidade decresce.

Na primeira etapa, quando a anotação é automática, todos os genes são anotados com uso de diversas ferramentas de bioinformática, sem interferência humana. Estas ferramentas buscam informações sobre o gene em diversas fontes de dados públicas, armazenam as informações obtidas na base de dados do organismo e atribuem automaticamente uma função ao gene conforme a ferramenta foi programada, levando em consideração principalmente a similaridade entre os genes. Na segunda etapa, na anotação manual, a anotação automática é revisada por um especialista denominado curador, que verifica as informações associadas automaticamente ao gene e, juntando ao seu próprio conhecimento que têm a respeito do organismo que está sendo anotado, atribui uma função ao gene.

O curador, ao atribuir função a um gene, além de considerar o fator similaridade e as informações armazenadas automaticamente, considera principalmente o conhecimento que tem no organismo que está anotando, sobretudo de fisiologia e bioquímica. Ao anotar, o curador faz uso também de uma grande quantidade de literatura (artigos) relacionada aos genes similares encontrados, a fim de pesquisar informações de relevância usadas na anotação dos genes. Após uma anotação, nem sempre fica explícito junto aos dados do gene anotado as evidências ou os critérios usados pelo curador para determinar a função do gene, não ficando claro o julgamento feito pelo anotador ao determinar a função do gene.

A falta de evidência seria simplesmente um problema local, pertinente somente ao genoma do organismo anotado, caso estes genes anotados não fossem disponibilizados em bases de dados públicas, as quais outros pesquisadores utilizam para anotarem o genoma dos organismos que pesquisam. A qualidade da anotação de um projeto genoma influencia na qualidade de futuros projetos genoma e, um projeto genoma atual é influenciado pela qualidade da anotação de projetos genoma anteriores. O depósito de novas anotações nas bases de dados públicas é diário, fazendo com que o volume de informação armazenada cresça rapidamente. Assim, a falta de evidência deixa de ser um problema local, pertinente somente ao genoma do organismo anotado, e se propaga para as anotações subseqüentes.

Com o grande volume de informação sendo depositado diariamente nos bancos genéticos espalhados pelo mundo, atingiu-se em poucos anos a quantidade de bilhões de pares de bases (pb) anotados; fato mencionado por Brudno et al. (2003) afirmando que em 2003 o volume de informações ainda era crescente e, por Rogozin et al. (2004) ao citar o rápido crescimento no número e diversidade dos

genomas procarióticos seqüenciados. O volume de dados continuou a crescer exponencialmente, como publicado em 2006 por Kulikova em seu trabalho sobre a base de dados genética mundialmente conhecida, o *Nucleotide Sequence Database*[1] *do European Molecular Biology Laboratory* (EMBL*)*, o qual cresceu de 58,7 milhões de setembro de 2005 para 80,5 milhões em setembro de 2006. Atualmente, existem milhões de genes de milhares de organismos pesquisados e anotados em diversos centros de pesquisa por inúmeros pesquisadores, armazenados em centenas de bases de dados genéticas distribuídas em todo o mundo.

Com este grande volume de informações genéticas, qualquer ferramenta que (1) ajude o curador a tomar suas decisões durante a anotação gênica, (2) que torne evidente os critérios usados pelos anotadores ao determinarem a função de um gene, (3) que direcione o anotador na seleção de literatura que possua informação mais relevante sobre um gene, ou (4) que minimize a quantidade de informação a ser lida pelo anotador, contribuem para o processo de anotação gênica reduzindo o tempo gasto pela equipe de anotadores e conseqüentemente os custos envolvidos no projeto.

Diariamente novos genomas são anotados e mais informações são depositadas nas bases de dados públicas, obrigando o anotador a pesquisar mais dados para decidir qual a função de um gene, quando esta função existir. Não raramente um anotador necessita ler dezenas de artigos em busca de informações que o ajude a tomar uma decisão sobre a função provável do gene. Uma ferramenta que faça a pré-leitura destas informações e selecione os artigos que revelem alguma informação que possa ajudar o anotador a decidir no momento de fazer a anotação gênica, se torna importante.

O genoma da bactéria *Chromobacterium violaceum* é um exemplo de anotação gênica feita no Brasil e disponível publicamente. O genoma da bactéria *C. violaceum* foi seqüenciado e anotado com uso da plataforma de anotação denominada *System for Automated Bacterial Integrated Annotation* (SABIÁ). Na

[1] http://www.ebi.ac.uk/embl

plataforma SABIÁ, após a anotação automática do genoma de *C. violaceum*, os dados foram revisados por curadores, os quais validaram a anotação automática e a atribuição de função aos genes. Para validarem a anotação automática, os curadores usaram quatro categorias pré-definidas para classificar os genes: c*onserved hypothetical, hypothetical, not valid e valid,* sendo utilizado também o campo de livre contexto *Notepad* para registrar o resultado do BLAST que determinou a atribuição de função ao gene.

Neste trabalho foi construido um software que acessa os diversos dados gerados pela anotação automática disponíveis ao anotador e insere neste conjunto de dados códigos de evidência, de forma a revelar ao leitor (anotador ou curador) a lógica pela qual uma função foi atribuída a um gene anotado no passado (Gb), fornecendo mais segurança ao anotador para decidir sobre a função do gene que está sendo anotado no momento (Gi), ou para que ele escolha, conforme as evidências apresentadas, a função de maior relevância ao gene. Para efeito de validação desta ferramenta será usado o genoma da bactéria *C. violaceum* previamente anotado.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver um software que, após a anotação automática, inclua aos dados do gene, códigos de evidência pré-definidos, proporcionando ao validador do gene, informações que revelem como os genes similares foram caracterizados e o auxilie a tomar uma decisão sobre sua validade.

1.1.2 Objetivos Específicos

Para atingir o objetivo geral, os seguintes objetivos específicos foram cumpridos:

- codificar um módulo de software que faça *download* e armazene em base de dados local a literatura referente aos genes válidos de *C. violaceum* anotados com uso da plataforma SABIÁ;
- registrar em base de dados local os códigos de evidência, os

termos relacionados a cada código de evidência e as regras relacionadas a cada código de evidência;

- codificar um módulo de software que compare os termos e regras registrados na base de dados com a literatura armazenada e determine automaticamente os códigos de evidência correspondentes aos genes válidos de *C. violaceum*;
- validar os códigos de evidência atribuídos pelo software a fim de verificar se revelam corretamente as informações encontradas na literatura;
- agregar valor às informações anotadas de *C. violaceum;*
- fornecer subsídio para a re-anotação do genoma de *C. violaceum* e anotação de futuros genomas.

2 REVISÃO BIBLIOGRÁFICA

Os primeiros relatos existentes referente às descobertas genéticas são de 1863, quando o monge agostiniano Gregor Mendel pesquisando ervilhas descobriu o mecanismo de herança através das células (OLIVEIRA; SANTOS; BELTRAMINI, 2004, p. 3). Mais tarde, em 1931, Phoebus Aaron e Theodor Levene identificaram os componentes básicos genéticos (bases nitrogenadas, açúcar e fosfato), conhecidos como nucleotídeos. Em 1953 houve outra grande descoberta genética que possibilitou a Francis Crick e James Watson ganharem um Nobel ao criarem o modelo químico de dupla hélice do ácido desoxirribonucléico (DNA) (OLIVEIRA; SANTOS; BELTRAMINI, 2004; SCHWARTZ, 2001).

Desde o ano de 1863 até aproximadamente 1990 muitas outras descobertas e avanços genéticos são relatados. A partir de 1990, a popularização dos bancos de dados (BD's), internet e novos recursos tecnológicos contribuíram para o crescimento e a disseminação das pesquisas genéticas existentes, possibilitando a criação de vários projetos genoma colaborativos, o rápido compartilhamento das informações entre os pesquisadores e a criação dos diversos bancos de dados biológicos (SILBERSCHATZ; KORTH; SUDARSHAN; 2006).

2.1 PROJETOS GENOMA

O conhecimento da seqüência do DNA é de grande interesse aos biólogos, uma vez que permite a dedução da presença de genes, e em última análise, da função desses genes para o organismo. Como conseqüência, o conhecimento do repertório completo dos genes de determinado organismo traz contribuição fundamental para o entendimento dos diversos processos biológicos que caracterizam aquele organismo. Com objetivo de conhecer o organismo humano, em 1990 foi estabelecido o Projeto Genoma Humano (PGH) nos Estados Unidos da América (EUA), com a finalidade de seqüenciar o genoma humano (LANDER et al., 2001). No bojo desse projeto estava incluído o desenvolvimento de técnicas

automatizadas de seqüenciamento, bem como de técnicas computacionais (bioinformática) que permitissem a manipulação da quantidade extraordinária de dados que seria gerada. Originalmente uma proposta norte-americana, o PGH se tornou uma iniciativa internacional com a adesão do Reino Unido, França, Alemanha, Japão e China, com um aporte de recursos jamais vistos em projetos biológicos. Apesar de considerado concluído em 2003, ainda hoje esforços são direcionados para entender melhor e caracterizar o genoma humano (The ENCODE Project Consortium, 2007), bem como de diversos outros organismos.

Enquanto o PGH era executado no exterior, o Brasil iniciava em 1997 as pesquisas genômicas, seqüenciando o genoma da bactéria *Xylella fastidiosa*. Para a execução desse projeto genoma foi organizada uma rede de 34 laboratórios biológicos distribuídos no estado de São Paulo e um centro de bioinformática situado na Universidade Estadual de Campinas (Unicamp), formando a rede ONSA (*Organization for Nucleotide Sequencing and Analysis)*, dando condições a cada pesquisador nos diferentes laboratórios para submeter e compartilhar as informações de um mesmo organismo (SIMPSON et al., 2000).

O projeto foi iniciado em 1997 e concluído em 2000, com a anotação de 2.838 genes, dando o devido reconhecimento ao Brasil pelas pesquisas genéticas, possibilitando sua participação no PGH, sendo o único país fora do eixo Estados Unidos, Europa e Japão a concluir o projeto genoma completo de um organismo (CAMARGO; SIMPSON, 2003).

Devido ao sucesso obtido com o genoma da *Xylella fastidiosa* e a importância deste conhecimento aplicado à economia agrícola do país, vários outros projetos genoma de outros patógenos de culturas nacional foram iniciados, como por exemplo: *Xanthomonas citrii*, *X. campestris, Leifsonia xyli* subsp. *xyli* e outros (CARRARO; KITAJIMA, 2002). Até 2003 havia sido completamente seqüenciado o genoma de dez organismos e mais doze projetos genoma estavam em progresso (CAMARGO; SIMPSON, 2003).

O sucesso da rede paulista de seqüenciamento de genomas fez aumentar o interesse na área genética no país e como conseqüência aumentaram também os investimentos. Estes recursos possibilitaram a criação de vários outros projetos genoma com estruturas tecnológicas similares, como o Projeto Genoma Brasileiro – Rede Nacional de Seqüenciamento (BRGENE), financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), e diversas redes regionais

como Projeto Genoma Sul (PIGS) e o Projeto Genoma do Paraná (Genopar), com recursos de órgãos de fomento estaduais, em parceira com o CNPq.

O Laboratório de Bioinformática (LABINFO) do Laboratório Nacional de Computação Científica (LNCC) de Petrópolis, no estado do RJ, foi o responsável pelo desenvolvimento da plataforma de anotação denominada SABIÁ, a qual será utilizada neste trabalho (ALMEIDA et al., 2004a; SIMPSON; CABALLERO, 2000). O primeiro organismo seqüenciado pela Rede Nacional de Seqüenciamento com uso da plataforma de anotação SABIÁ foi a bactéria *Chromobacterium violaceum* (VASCONCELOS et al., 2003), sendo que seus 4.431 genes foram seqüenciados e anotados de maneira colaborativa, distribuídos entre os 25 laboratórios participantes.

2.2 PLATAFORMA SABIÁ

A plataforma SABIÁ (*System for Automated Bacterial Integrated Annotation*)[2] foi desenvolvida para preencher as necessidades do Projeto Genoma Brasileiro no gerenciamento, montagem e anotação do genoma da bactéria *Chromobacterium violaceum* (ALMEIDA et al., 2004b). A Figura 1 mostra parcialmente uma página de anotação, a qual está disponível para ser acessada via internet junto com todo o projeto SABIÁ.

Segundo Vasconcelos et al. (2003), o seqüenciamento e análise do genoma da *C. violaceum* foi executado por 25 laboratórios, um centro de bioinformática e três laboratórios de coordenação distribuídos pelo Brasil. Embora não tenha sido o primeiro genoma a ser seqüenciado no Brasil, foi o primeiro executado em escala nacional.

[2] http://www.SABIÁ.lncc.br/

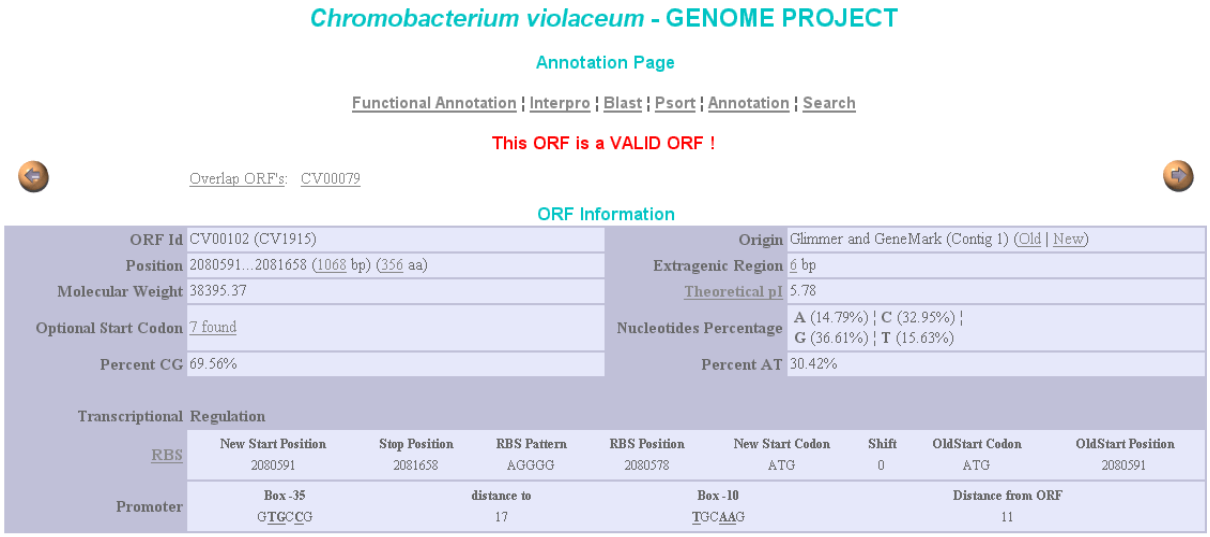


Figura 1 - Tela parcial de anotação da plataforma SABIÁ

A plataforma SABIÁ é constituída por vários programas escritos na linguagem de programação PERL (*Practical Extraction and Report Language*), os quais acessam uma base de dados implementada no sistema de gerenciamento de banco de dados (SGBD) MySQL e é executado por meio do protocolo *hyper text tranfer protocol* (http) do servidor Apache (ALMEIDA et al., 2004a). As instruções para instalação e os módulos da plataforma SABIÁ podem ser adquiridos gratuitamente e instalados após aprovação da licença de uso que pode ser obtida na página do projeto[3].

Executado por meio de interface web, o SABIÁ permite a submissão e pesquisa de dados, bem como interage com vários programas, como: Glimmer, Genemark, tRNAScan-SE, coleção de programas da família BLAST (*Basic Local Alignment Search Tool* - blastn, blastp, blastx, tblastn e tblastx), Psort, InterPro, KEGG, COG, phred, phrap, GO, RBSFinder, PSORT (ALMEIDA et al., 2004a).

Mesmo os pesquisadores integrando um mesmo projeto, concentrando a análise por bioinformática no LABINFO do LNCC e, tendo feito treinamento dos anotadores, a anotação gênica não foi completamente padronizada. Este fato permitiu que vários anotadores informassem o resultado de suas anotações

[3] Licença de uso: http://www.SABIÁ.lncc.br/SABIAport.pdf

conforme preferência individual, resultando em uma base de dados sem um padrão de anotação para alguns campos, como o "*Notepad*" da Figura 2. Um subgrupo de anotadores fez uma revisão final de todas as anotações feitas pelo conjunto total de anotadores, mas mesmo dentro desse subgrupo não houve esforço em padronizar a anotação manual.

Sobre este modelo de anotação distribuído, onde a anotação gênica é feita por anotadores em vários pontos geográficos, Elsik et al. (2006) apontou algumas vantagens, alertando da "necessidade de manter consistente a qualidade, apesar da diversidade de perícia em anotação na comunidade; manter consistente o formato de dados; e minimizar o potencial em anotações duplicadas", destacando principalmente esforços direcionados em anotações duplicadas e a utilização de diferentes padrões e formas de apresentação dos dados, aspectos reforçados por Claverie (2000).

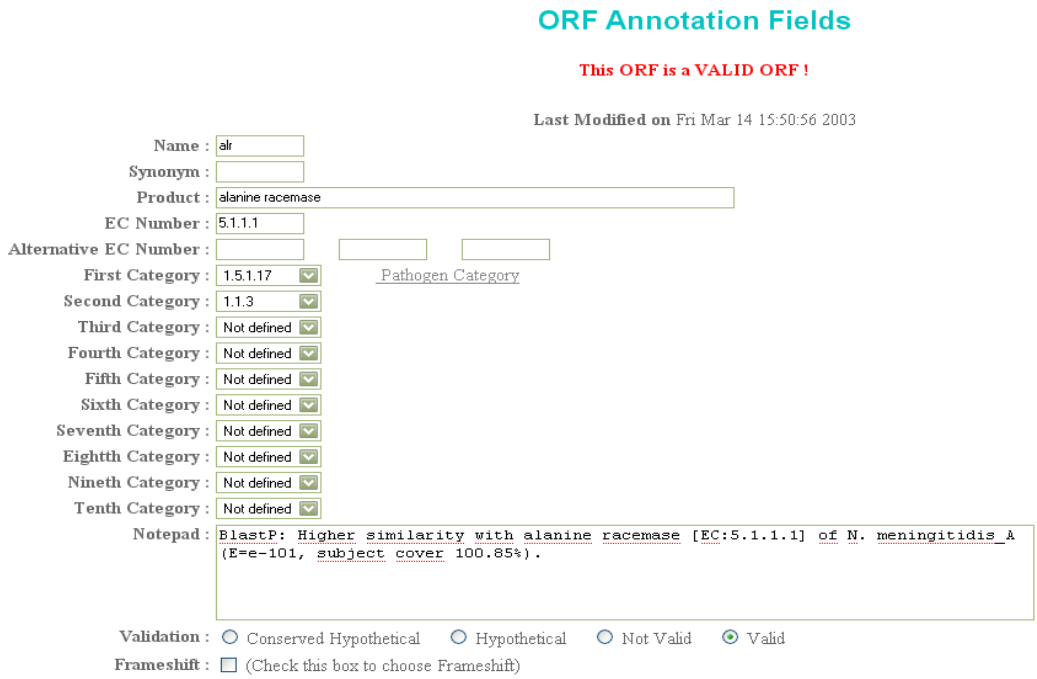


Figura 2 - Tela parcial do anotador na plataforma SABIÁ, destacando campos de anotação manual.

O problema na qualidade da informação e da falta de padrões para descrição de genes e seus produtos em uma plataforma de anotação ou em diferentes bancos de dados podem ser minimizados, senão resolvido, com o uso de ontologias. Segundo Gruber (1993), "ontologia é uma especificação explícita de uma conceitualização", onde entende-se "conceitualização" como um fenômeno abstraído do mundo real, de forma a identificar seus conceitos relevantes, e "explícito" como a definição prévia dos conceitos a serem utilizados e das restrições a serem aplicadas.

Com a intenção de melhor integrar os projetos genoma para que as pesquisas e dados do seqüenciamento pudessem ser aproveitados por outros projetos genoma e principalmente compreendidos pelos pesquisadores, houve esforços para padronização na forma de descrever os dados genéticos, sendo que o Gene Ontology (GO) tem se destacado.

2.3 GENE ONTOLOGY

O projeto Gene Ontology (GO) é um esforço colaborativo para direcionar a necessidade por descrições consistentes de produtos genéticos em diferentes bases de dados. O projeto começou com a colaboração entre três modelos de organismos, o Flybase, o *Saccharomyces Genome Database* (SGD) e o *Mouse Genome Database* (MGD) em 1998. Desde então o consórcio GO tem crescido para incluir bancos de dados de diversos organismos. No site é possível ter acesso às ontologias bem como fazer *download*[4] livremente de várias bases de dados genéticos de diferentes organismos (The Gene Ontology, 2008).

Segundo Liu, Hu e Wu (2005) “o desenvolvimento do GO como um vocabulário comum para anotação, permite integrar consultas cruzando múltiplas bases de dados e identificar genes relatados semanticamente e produtos genéticos”, desde que os projetos genoma tenham os termos GO associados durante a anotação. Como afirmado por Neerincx e Launissen (2005), com o crescente número de ferramentas e bases de dados, se torna cada vez mais necessário que estas se comuniquem e o GO, com uso de ontologias, viabiliza esta integração. Esta integração é defendida por Rogozin et al. (2004), ao citar que as bases de dados devem combinar diferentes aspectos no contexto da genômica para uma segura predição de associações funcionais aos genes.

As ontologias do GO estão divididas em três categorias: (1) processo biológico (BP), que se refere a um objetivo biológico para qual o gene ou seu produto contribui; (2) função molecular (MF), definida como a atividade bioquímica

[4] http://www.geneontology.org/GO.current.annotations.shtml

do produto de um gene; e (3) componente celular (CC), que se refere ao lugar na célula onde o produto gênico está ativo (ASHBURNER et al., 2000). Estas categorias praticamente respondem as respectivas questões: “o que são?”, “como desencadeiam” e “onde ocorrem?” os processos celulares.

Definido pelo consórcio GO, existe algumas informações que são requeridas para o processo de anotação: (1) nome do gene, termo associado e seu ID; (2) código de evidência, que identifica como o produto do gene foi caracterizado; e (3) referência, data de anotação e base de dados ou o responsável pela anotação (BERARDINI et al., 2004). Estas informações são fornecidas pelo anotador no momento da anotação gênica quando a plataforma de anotação estiver preparada para trabalhar com as ontologias definidas no GO.

Para o desenvolvimento deste trabalho, foram inseridos na anotação do genoma de *C. violaceum,* os códigos de evidência definidos pelo GO que revelam informação sobre o processo usado na anotação gênica e que indicam como a função de um gene foi provavelmente determinada.

2.4 ANOTAÇÃO GÊNICA

O processo de anotação gênica se inicia após a etapa de seqüenciamento e montagem do genoma, quando é determinada a seqüência de bases adenina (A), citosina (C), timina (T) ou guanina (G) do DNA de um organismo. Dentro desta grande cadeia de caracteres que forma o DNA de um organismo, se encontram o(s) gene(s) e estão inseridos entre uma seqüência de bases que marca o início (normalmente ATG) do gene e uma seqüência de bases que marca o término (TAA ou TAG ou TGA) do gene. Cada três pares de bases (códon) codificam um aminoácido. Uma cadeia (seqüência) de aminoácidos codifica uma determinada proteína. Com quatro bases (A, C, T ou G) e três posições para formar um códon, é possível formar 64 (4^3) códons. Existe, porém, somente 20 aminoácidos, levando a conclusão que um aminoácido pode ser representado por mais de um códon (BERGERON, 2002).

O processo de anotação gênica, além de ter a finalidade de encontrar os genes, consiste em confrontar cada gene (cadeia de caracteres) com genes em outras bases de dados genéticas existentes (Genbank[5] e Swiss-Prot[6] por exemplo), comparando a similaridade encontrada entre as seqüências dos genes de diversos organismos (CARRARO; KITAJIMA, 2002). Se o gene de um determinado organismo for similar aos genes de outros organismos, existe a possibilidade de que esses genes similares possuam um ancestral comum, ou seja, de que sejam genes homólogos, situação onde se deduz que o gene estudado tem a mesma função biológica do outro gene disponibilizado em base de dados pública e com função biológica já conhecida. Genes homólogos estão divididos em duas classes: ortólogos e parólogos. Dois genes são ortólogos quando pertencem a espécies distintas. Dois genes são parólogos quando a duplicação do gene, seguida de mutações, ocorreu na própria espécie, gerando outro gene.

Existem vários programas que podem ser utilizados para fazer o processo de anotação gênica, alguns comercializados sob licença livre, como: GenDB, MAGPIE, Pedant, GeneQuiz, Artemis e Blastx (MEYERS et al., 2003), (LLIOPOULOS et al., 2003); e outros sob licença comercial, como ERGO, Pedant-Pro, Phylosofer, BioScout, WIT (MEYERS et al., 2003). As anotações poderiam ser executadas de forma totalmente automática, sem a necessidade de especialistas, caso somente a similaridade fosse importante.

A anotação gênica para ser válida não depende somente do grau de similaridade encontrado entre os genes, mas depende também do fato dos genes serem ou não homólogos, depende do conhecimento que um anotador tem sobre o organismo e de como este organismo opera. “Para fazer um julgamento de confiança, os curadores precisam de um grande volume de informação em uma ordem estruturada apropriada que os permitam tomar uma decisão em cada estágio do processo” (KASUKAWA et al., 2003).

Também é possível realizar anotação baseada na caracterização experimental das funções dos genes, processo executado em laboratório que é

[5] http://www.ncbi.nih.gov/
[6] Base de dados localizada no servidor Expasy, Switzerland - http://www.expasy.org/sprot/

demorado e de alto custo, porém, gera informações confiáveis sobre a função de um gene. Supondo que um gene (Gp) tenha passado por um processo de caracterização experimental e suas informações estejam disponíveis em um BD público, em um processo de anotação, caso seja encontrado um gene (Gi) com elevada similaridade em relação ao gene caracterizado (Gb), poderá ser atribuído ao gene encontrado (Gi) a função de seu gene homólogo com caracterização experimental (Gb), pois têm a mesma função. Esta é a base para anotações automáticas por similaridade genética, onde se atribui uma função a um gene (Gi) baseado na função de outro gene já caracterizado (Gb).

Esta busca e comparação de similaridade pode ser feita por dois processos: (1) anotação manual, onde um especialista procura características em outros genes ou, (2) por anotação automática, onde existe um conjunto de software para procurar características em outros genes. O processo de anotação gênica pode ser totalmente manual, totalmente automático ou uma combinação de ambos, onde após a anotação automática ser concluída, é revisada por um curador. Um dos softwares mais utilizados no processo de anotação gênica para buscar genes similares é o BLAST, que foi utilizado para fazer a anotação gênica da bactéria *C. violaceum* na plataforma SABIÁ.

2.4.1 Anotação Manual *vs* Anotação Automática

Toda anotação gênica feita por software é denominada “automática”. Porém, quando o resultado desta anotação é verificado e validado por um especialista, conhecido como curador, esta anotação passa a ser denominada de “anotação manual”. Há muita discussão entre estas duas formas de anotação pelos autores das áreas de genética e bioinformática.

Com o grande volume de informações e quantidade de genomas seqüenciados constantemente, seria inadmissível não esperar o uso de recursos de bioinformática ou prover um alto nível de exame e correção manual para descrever precisamente os genes presentes nestes genomas (MUNGALL et al., 2002), sendo necessário tanto a anotação automática quanto a manual.

Bailey et al. (1998), ao descrever o software de anotação automática Genome Annotation and Information Analysis (GAIA), destacou como um diferencial ao descrever um processo comparativo e automático para iniciar a caracterização genética, sem comprovação experimental dos genes em laboratório:

> A necessidade de métodos de elevado desempenho com base mais na predição computacional do que na descrição de resultados de laboratório... e uma ênfase em consistência e confiabilidade da anotação com uso de grandes quantidades de seqüência.

Kasukawa et al. (2003) complementa que “embora nós tenhamos formidável confiança na anotação automática, compreendemos que existirá ainda a necessidade por revisão e verificação adicional humana para avaliar e confirmar esta tarefa”. Destacando ainda que “uma análise no padrão humano de verificação sugere que um bom projeto preliminar de anotação pode evitar a necessidade de intervenção humana em muitos casos”. Todavia, esta revisão não precisa ser feita na totalidade dos genes, pois em um trabalho a anotação automática foi aceita em mais de 85% dos casos, sendo clara a necessidade de ambas as formas de anotação, uma automática preliminar e um especialista humano para rever e validar a tarefa inicial. Desta forma seria anotado um grande volume de genes automaticamente e revisados pelos especialistas algumas informações mais significativas.

Considerando que o processo de anotação depende de informações genéticas previamente anotadas e armazenadas em bases de dados públicas, o processo de anotação e validação dos dados depende muito de curadores treinados e da equipe de anotação que asseguram que a informação recebida é completa e exata (BOURNE; McENTYRE, 2006).

Bryson et al. (2006) discute características que estas ferramentas devem possuir, dizendo que “os biologistas necessitam examinar a análise feita de forma automática e, combinado com seus próprios conhecimentos do organismo, formar uma decisão global sobre a natureza dos diferentes elementos genômicos, como o gene”. Como ocorre com a base de dados de *C. violaceum,* já anotada e revisada na plataforma de anotação SABIÁ, complementa Lewis et al. (2002):

> A comunidade genômica requer ferramentas que disponibilizaria mais que uma simples visão dos dados, mas que habilitaria curadores profissionais, e finalmente cada pesquisador, para facilmente modificar e corrigir diretamente as anotações sobre uma base de dados.

Kasukawa et al. (2003) destaca três características consideradas ideais para a combinação de anotação manual e automática em um sistema, já que são totalmente interligadas: (1) capacidade de apontar as anotações geradas automaticamente e rejeitadas, as quais poderiam ser recolhidas e transformadas numa nova fonte a ser refinada; (2) capacidade para determinar se uma anotação

necessita ou não ser verificada manualmente; e (3) capacidade para detectar as anotações que podem ser alteradas em uma atualização na base de dados.

Prlic et al. (2004) destaca outros quatro recursos também importantes para as ferramentas de anotação automática:

> (1) a integração de vários tipos de dados biológicos derivados de fontes diferentes, (2) o gerenciamento de grandes volumes de informações, (3) atualização freqüente de bases de dados públicas, e (4) ser extensível para a integração de novas fontes de dados.

A anotação automática é fortemente utilizada devido à grande quantidade de dados que é analisada ao se fazer anotação gênica, não sendo dispensada porém a anotação manual, pois os dados anotados automaticamente precisam ser revisados por um especialista. Uma característica importante para um software de anotação é informar ao curador o que deve ser revisado ou mesmo classificar a anotação automática de forma que a anotação manual não se torne necessária em todo o genoma, poupando tempo aos curadores. Como novos genomas seqüenciados são depositados e alterados diariamente nas bases de dados públicas, outra boa característica para um software de anotação automática seria a de “perceber” e “informar” ao usuário quais as mudanças e quais genes foram alterados desde a última anotação, evitando ter que re-anotar os genomas constantemente para que permaneçam atualizados.

2.4.2 Re-anotação

Re-anotação quer dizer anotar novamente um genoma já anotado no passado e deveria ser um processo contínuo. Como novos organismos são seqüenciados constantemente e mais genes são armazenados nas bases de dados públicas, um genoma anotado anteriormente pode ficar desatualizado, pois, dentre estes novos genes inseridos, alguns poderiam ser homólogos ao gene de um organismo anteriormente anotado e que poderia ser atualizado. Segundo Srdanovic et al. (2005), “sob o ponto de vista do usuário, estes recursos são mais úteis quando são atualizados regularmente...”, considerando que o usuário a quem o autor se refere é um anotador, alguém que faz uso dos genes armazenados nas bases de dados públicas que poderiam estar com informações atualizadas caso fossem re-anotados. O custo envolvido na anotação, ou re-anotação, é um fator que desmotiva a re-anotação completa e continuada dos genomas.

Kersey et al. (2005), ao relatar o desenvolvimento da ferramenta para re-anotação denominada Integr8 afirmou que, “como a anotação para predição de genes é às vezes inferida por similaridade com outras seqüências, informações podem se tornar desatualizadas pela submissão de entradas adicionais na base de dados”. No momento de fazer a anotação gênica, as bases de dados públicas são verificadas a fim de encontrar genes homólogos com o gene que se deseja anotar e o serviço só termina após todos os genes do organismo serem anotados. Em tempo paralelo a esta anotação, ou posteriormente, outros biólogos em outros centros de pesquisa espalhados pelo mundo estão fazendo anotações de outros organismos e depositando novos genes nestas bases de dados. Por este motivo, uma anotação gênica feita no passado fica desatualizada, pois, após a anotação, podem ocorrer inserções de genes que tenham homologia com os genes do organismo já anotado e que não foram comparados por não estarem na base de dados no momento da anotação. Ao fazer uma re-anotação, toda a base de dados é verificada novamente, garantindo que os novos genes inseridos fossem comparados.

Re-anotações são importantes também para manter uma anotação gênica atualizada e livre de erros, pois não são raras as alterações realizadas nos genomas das diversas bases de dados ocasionadas por erros de anotação e, estas alterações podem ter influência sobre uma anotação feita no passado e não atualizada (NIELSEN; KROGH, 2005), (VELOSO, 2005).

Re-anotação se torna importante ainda para aumentar o conhecimento sobre genética. Segundo Kasukawa et al. (2003):

> Baseado em uma compreensão de como anotadores humanos trabalham, acreditamos que anotação computacional completa e contínua atualização de seqüências serão possíveis e que isto pode conduzir a recursos muito mais dinâmicos para o estudo da genômica funcional.

Weinel, Ermolaeva e Ouzounis (2003) acrescentam:

> Uma vez que um projeto genoma é terminado e liberado, sua futura re-anotação é a princípio mantido pela respectiva comunidade de pesquisa que se mantém a par de atualizar, corrigir e melhorar constantemente a evidência funcional para os produtos do gene. Uma re-anotação rotineiramente automática pode assim fornecer uma base boa para esta melhoria contínua do conhecimento sobre as funções codificadas em um genoma.

Porém, re-anotar não é uma tarefa simples. Com o grande volume de informações não se pode esperar que tudo seja examinado e corrigido manualmente e com grande precisão, pois isto consumiria muito tempo e geraria custos. Por outro

lado, fazer todo o processo automaticamente pode conduzir a erros de anotação, pois é necessário o auxílio de um especialista para inferir qualidade aos dados (MUNGALL et al., 2002). Neste sentido, ferramentas que (1) ajudem os especialistas a revisarem os dados, (2) que consigam fazer a anotação com precisão ou mesmo (3) que disponibilizem mais informações para auxiliar um especialista (ou outra ferramenta) a fazer um julgamento correto são importantes e necessárias.

Um exemplo de sucesso em atualização contínua de genoma e que mostra a importância de se re-anotar é dado por Winsor et al. (2005) ao descrever a evolução e atualização do genoma da bactéria *Pseudomonas aeruginosa*. Segundo o autor o número de pesquisadores envolvidos na anotação partiu de 61 no ano de 2000 para 105 até julho de 2004; distribuídos em 11 países contribuíram com a atualização de 1019 anotações, sem contar com as submissões feitas antes da publicação original do artigo escrito por Stover et al. (2000).

Este trabalho está inserido no contexto da re-anotação ao computar a literatura usada na anotação do genoma de *C. violaceum* e atribuir os códigos de evidência encontrados nesta literatura, o que possibilita reduzir o tempo necessário para a re-anotação de um genoma ao direcionar os curadores à leitura de informação previamente classificada e relevante para a atribuição de função aos genes (PÉREZ et al., 2004). Está também inserido no contexto de re-anotação para outros genomas, possibilitando que outros anotadores, usuários das informações de *C. violaceum*, conheçam em que foi provavelmente fundamentada a atribuição de função a cada gene e saibam quais critérios cada anotador usou para tomar sua decisão, dando confiabilidade à anotação gênica de *C. violaceum*.

2.4.3 Importância de Anotação Gênica Correta

Scott (2006) enfatiza a importância de boas anotações ao dizer:

> Estudos evolucionários fornecem caminhos para a descoberta de mais alvos para drogas, o desenvolvimento de medicina personalizada, culturas resistentes a inseto, bactérias para descontaminar ambientes poluídos, entre outros. Estes tipos de análises e de muitas outras não podem ser executadas se os genomas forem mal anotados.

Não importa que seja usado um processo de anotação manual, automática ou mista, o importante é que as informações atribuídas a um gene sejam corretas. Para fazer anotação do gene de interesse (Gi), o biólogo depende de anotações pré-existentes em alguma base de dados, porém, depende também que a anotação dos

genes existentes nestas bases (Gb) esteja correta. Caso as anotações não estejam corretas e o biólogo não perceba, a sua própria anotação conterá erros, induzindo futuros pesquisadores e anotadores, usuários desta anotação (Gi), ao erro. Normalmente, para evitar que erros assim ocorram, curadores verificam os dados anotados, analisam e comparam as informações com outras bases de dados para determinar o grau de confiança da anotação.

Elsik et al. (2006) aponta como um dos aspectos mais desafiadores na tarefa de anotação (1) manter a qualidade apesar da diversidade de peritos em anotação, acrescentando ainda como fator importante (2) manter formatos de dados consistentes e (3) minimizar o potencial para a anotação duplicada. Considerando o primeiro aspecto apontado por Elsik et al. (2006), mesmo os curadores verificando o genoma e inferindo funções acertadas aos genes, cada um o faz conforme regras do projeto e os próprios conhecimentos, nem sempre revelando o que considerou ao decidir que uma anotação está correta.

Softwares usados na anotação gênica por anotadores de diversos países, como o BLAST, fazem busca por genes similares anotados e trazem junto a cada gene encontrado, dados que podem revelar a qualidade e o processo de anotação realizado para cada gene. Mesmo existindo esta informação, nem sempre ela é expressa pelos anotadores ou mesmo não está prevista nos projetos genoma a inserção de tais informações, não ficando claro a futuros usuários desta informação o que motivou o anotador a atribuir uma função ao gene.

2.4.4 Ferramenta de Anotação BLAST

O BLAST (*Basic Local Alignment Search Tool*) foi desenvolvido no *National Center for Biotechnology Information* (NCBI) e é a ferramenta de bioinformática mais usada dentre todas as desenvolvidas para fazer busca por similaridade em seqüências biológicas (PERTSEMLIDIS; FONDON, 2001).

A família de algoritmos BLAST foi utilizada para fazer anotação gênica na plataforma SABIÁ e permitem a comparação de similaridade entre seqüências de nucleotídeos (BlastN) ou proteínas (BlastP). Dado uma seqüência genética, o algoritmo a compara com outras seqüências depositadas em outros bancos de dados públicos e retorna um conjunto de seqüências genéticas (Figura 3b) classificada em ordem decrescente de similaridade.

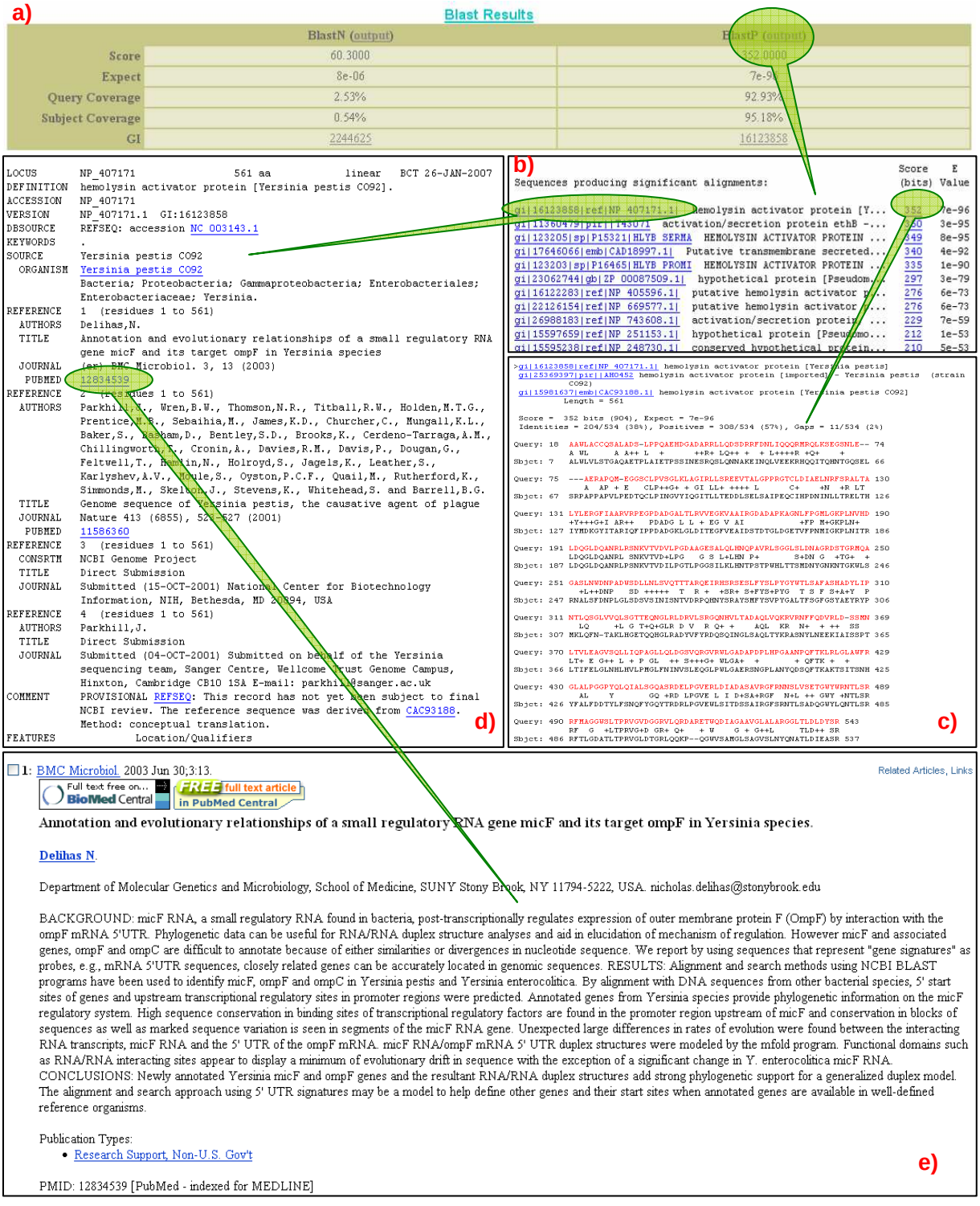


Figura 3 – Anotação CV00068 armazenada no SABIÁ. a) Tela do SABIÁ que fornece um link ao resultado BLAST armazenado no SABIÁ b) Resultado BLAST armazenado no SABIÁ com as seqüências mais significantes. c) Detalhes da comparação da similaridade dos genes fornecidos pelo BLAST. d) Relação de artigos publicados no NCBI referente a um dos resultados do BLAST com link para o resumo do artigos no PubMed e) Exemplo do resumo do um artigo armazenado no PubMed acessado por um resultado do BLAST (b).

Após a execução do programa BLAST no genoma de *C. violaceum*, o resultado retornado foi armazenado na base de dados da plataforma SABIÁ (Figura

3a, 3b e 3c), permitindo aos anotadores validarem e registrarem suas observações (Figura 2) considerando as informações apresentadas. A página de resultados do BLAST foi armazenada no SABIÁ em formato *html* (*hyper text markup language*) (Figura 3c) e, possui *links* para uma relação de artigos depositados no Centro Nacional de Informação Biotecnológica (NCBI)[7] referente ao gene similar encontrado (Figura 3d). Desta relação de artigos é possível usar os links para navegar aos resumos de cada artigo (Figura 3e).

Ao fazer a anotação gênica, o anotador usa os resultados do BLAST para acessar a literatura disponível no NCBI sobre os genes similares encontrados. Nesta literatura estão contidas informações que podem revelar a forma como um gene foi caracterizado, fornecendo subsídios para o anotador ou curador atribuir uma função a um gene baseado em informações seguras.

2.4.5 Códigos de Evidência

Os códigos de evidência, ou categorias, são usados em projetos genoma para indicar o estado de uma anotação ou revelar a forma como a anotação está sendo executada. Não existe um padrão ou regra que deve ser seguida no uso dos códigos de evidência, nem mesmo é obrigatório adotar tal procedimento, cabendo a decisão de “se” e “como” usar os códigos de evidência a cada equipe ou projeto. Na seção 2.4.6 é feita uma revisão sobre a forma que os códigos de evidência são usados em alguns projetos genoma, os quais também recebem nomenclaturas como “*evidence code*”, “*confidence level*”, “categoria", “validação” ou simplesmente “*level*”. Neste trabalho usaremos o termo “código de evidência”.

Para o GO, um código de evidência indica como a anotação para um termo do GO em particular é suportada, não sendo necessariamente a classificação do tipo de experimento ou análise feita (The Gene Ontology, 2008). Berardini et al. (2004) fornece um exemplo de como os códigos de evidência podem ser usados ao fazer anotação gênica: “usando a combinação de termos GO e códigos de evidência, um

[7] http://www.ncbi.nlm.nih.gov

pesquisador visando um novo projeto pode ter uma visão atualizada dos genes que ainda requerem caracterização experimental".

Uma evidência também é definida como "um código que descreve o tipo de análise realizado junto com [o gene], em alguns casos, uma referência a outro objeto na base de dados que suporte a evidência" (Flybase – Reference Manual G, 2008).

Nos diversos projetos genoma, quando da existência de algum código de evidência, este pode aparecer como uma hierarquia de números, como por exemplo, de um a dez ou um a cinco, onde o significado de cada nível é particular a cada projeto. Os códigos de evidência podem também ser representados por uma letra, uma palavra ou uma frase.

Na plataforma SABIÁ, usada na anotação do genoma de *C. violaceum*, foram usadas as palavras "*valid", "conserved hypothetical", "hypothetical"* e *"invalid"* para definir as categorias dos genes (ALMEIDA et al., 2004a).

O GO, em um esforço de padronizar a forma de qualificar as anotações genéticas, criou dezoito códigos de evidência, os quais foram extraídos e traduzidos da página que descreve o projeto (The Gene Ontology, 2008). Estes códigos, assim como o seu respectivo significado são apresentados no Quadro 1.

Código de evidência	**Significado**
Inferred by Curator (IC)	Usado em casos onde uma anotação não é suportada por qualquer outro código de evidência, mas pode ser inferido por um curador de outra anotação GO, para a qual a evidência está disponível.
Inferred from Experiment (EXP)	Código usado para indicar que um experimento foi encontrado na literatura citada, cujos resultados indiquem uma função do gene, um processo ou uma localização subcelular. Este código somente deve ser usado quando os códigos de evidência IDA, IPI, IMP, IGI e IEP não possam ser usados.
Inferred from Direct Assay (IDA)	Usado para indicar um ensaio bioquímico, um experimento para determinar a função, processo ou componente de um gene indicado por um termo GO.
Inferred from Electronic Annotation (IEA)	Usado para anotação baseada em similaridade ou transferida de outra base de dados, desde que não tenha sido revista por um especialista.
Inferred from Expression Pattern (IEP)	Usado para anotação originada com base no conhecimento temporal ou geográfico (momento ou local) da expressão de um gene, quando se conhece quando ou onde uma expressão do gene ocorre.
Inferred from Genomic Context (IGC)	Este código de evidência pode ser utilizado sempre que informação sobre o contexto genômico de um produto gênico constitui-se parte da evidência para uma

	anotação específica. O contexto genômico inclui, mas não se limita, a aspectos como a identidade de genes vizinhos ao produto gênico em questão (isto é, sintenia), estrutura de operon, análise filogenética ou outra análise de genoma completo.
Inferred from Genetic Interaction (IGI)	Usado em todas as combinações de alteração na seqüência (mutações) ou na expressão de mais de um gene ou seu produto, desde que as mutações sejam documentadas. Usado também para situações onde a mutação de um gene (A) fornece informação sobre a função, processo ou componente de outro gene (B). O gene B é anotado com IGI.
Inferred from Mutant Phenotype (IMP)	Usado para qualificar tudo que é concluído pela observação de variações ou mudanças, como mutações ou níveis anormais do produto(s) de um único gene de interesse. Ideal que seja usado onde uma situação anormal prevalece na célula ou organismo devido às alterações da seqüência ou da expressão de um gene. Usado também para as experiências em que a inferência é baseada nos efeitos observados na ocorrência natural de variações em um gene.
Inferred from Physical Interaction (IPI)	Usado para cobrir interações físicas naturais entre o produto do gene de interesse e outra molécula.
Inferred from Sequence or Structural Similarity (ISS)	Usado para toda análise baseada em alinhamento de seqüência, comparação de estruturas ou avaliação das características da seqüência como sua composição, desde que revisado por um especialista.
Inferred from Sequence Orthology (ISO)	Subcategoria de ISS. Indica que a seqüência genética tem similaridade com outro gene em espécie diferente (ortólogo), indicando que os genes derivam de um ancestral comum.
Inferred from Sequence Alignment (ISA)	Subcategoria de ISS. Usado quando o alinhamento de seqüência é a base para fazer uma anotação. Deve ser usado somente se um curador revisou manualmente a evidência.
Inferred from Sequence Model (ISM)	Subcategoria de ISS. Usado quando um modelo estatístico é usado para fazer a predição sobre a função de uma proteína ou RNA de uma seqüência ou grupo de seqüência.
Non-traceable Author Statement (NAS)	Usado em casos onde a publicação que um especialista utiliza para dar suporte a uma anotação não mostra a evidência, não sendo possível referenciar o trabalho original. Usado para qualificar bases de dados que não citam os artigos e, indicações em artigos que um especialista não pode rastrear outra publicação para comprovar suas evidências.
No biological Data available (ND)	Usado em anotações onde a função molecular, processo biológico ou componente celular são desconhecidos.
Inferred from Reviewed	Usado quando os resultados de grandes seqüências de

Computational Analysis (RCA)	DNA forem comparados com os resultados de outros experimentos em larga-escala, como predição baseada na integração de séries de dados de diferentes tipos e predição baseada em extração de texto (*text mining*).
Traceable Author Statement (TAS)	Usado em casos onde a publicação que um especialista utiliza para dar suporte a uma anotação não mostra a evidência, sendo possível, porém, referenciar o trabalho original.
Not Recorded (NR)	Usado para anotações feitas antes que os especialistas começassem a rastrear os tipos de evidência. Aparecem em anotações legadas e não deve ser usado para novas anotações.

Quadro 1 – Códigos de evidência usados pelo GO.

O GO dividiu os códigos de evidência em cinco categorias: (1) códigos experimentais (EXP, IDA, IPI, IMP, IGI, IEP); (2) códigos de análise computacional (ISS, ISO, ISA, ISM, IGC e RCA); (3) códigos declarados por autor (TAS e NAS); (4) códigos declarados por curador (IC, ND) e (5) códigos definidos automaticamente (IEA) (The Gene Ontology, 2008). Os códigos de evidência EXP, ISO, ISA e ISM foram criados pelo GO ainda em 2008, sendo que no início deste trabalho eles não existiam. A qualquer momento novos códigos de evidência podem ser incluídos ou excluídos, haja vista que as regras de “se” e “como” utilizar os códigos de evidência são definidas em cada projeto.

Berardini et al. (2004) já enfatizava a importância de se usar os códigos de evidência dizendo que eles “podem ser usados para determinar a que extensão um gene foi caracterizado”. Saxonov, Berg e Brutlag (2006) usaram em seu trabalho de análise do genoma humano somente as anotações que tivessem os códigos de evidência IDA, IEP, IGI, IMP, IPI, ISS, e TAS.

Para a realização deste trabalho foram usados os códigos de evidência IDA, IEP, IGC, IGI, IMP, IPI e ISS para caracterizar o genoma da bactéria *C. violaceum* anotado na plataforma SABIÁ.

2.4.6 Códigos de Evidências em Projetos genoma

Os níveis de evidência, tais como existem hoje, não foram usados desde o início dos projetos genoma, quando surgiu o interesse em conhecer o funcionamento dos diversos organismos. Um dos maiores projetos genoma já realizado e revisado por curadores, o Projeto Genoma humano (HGP), não menciona o uso de níveis de

evidência e nem o uso de ontologias, o que é natural para qualquer tecnologia que acaba de surgir (*International Human Genome Sequencing Consortium*, 2004).

Um dos primeiros ensaios em direção à criação de ontologias e códigos de evidência foi criado por Bailey et al. (1998) ao relatar o uso de um vocabulário controlado para especificar as características dos genes anotados. No projeto de Bailey et al. (1998) as informações armazenadas no campo "*Origin*" eram preenchidas com palavras-chave que especificavam como os dados foram obtidos. O vocabulário era composto pelas palavras e significados apresentados no campo *"origin"* do Quadro 2.

Origin	**Significado**
Automatic	Usado quando os dados eram obtidos por anotação automática, usando um subsistema denominado CARTA.
Manual	Usado após os dados gerados automaticamente (*automatic)* serem validados por um curador. A validação era feita pelo sistema ATLAS pela contribuição humana direta.
GenBank	Quando os dados eram originados de uma descrição em um registro da base de dados *Genbank*.

Quadro 2 - Códigos de evidência usados por Bailey et al. (1998).

Takami et al. (2000) ao anotar o genoma da bactéria *Bacillus halodurans* e compará-lo com *Bacillus subtilis* usou como forma de evidenciar a anotação os códigos: (1) "*conserved",* para indicar a existência de similaridade entre os genes dos dois organismos e, dividindo o código "*conserved*" em (1.1) "função conhecida", quando uma função fosse encontrada para um gene e (1.2) "função desconhecida", quando nenhuma função fosse atribuída ao gene. Houve uso também do código (2) "*non-conserved"*, para indicar que não foi encontrada similaridade com outros genes.

Stover et al. (2000) ao seqüenciar o genoma da bactéria *Pseudomonas aeruginosa* fez uso de códigos de evidência para caracterizar o organismo usando numeração seqüencial de um a cinco, onde representou do maior nível de confiança para o menor, como pode ser observado no Quadro 3:

Nível	**Significado**
1	Genes com função previamente determinada para o mesmo organismo.
2	Forte homologia dos genes com função reconhecida em outros organismos.
3	Gene com função atribuída baseada nos resultados de busca de *motif* com homologia limitada com outros genes.
4	Gene com homologia com outros genes de função desconhecida.
5	Gene sem homologia com qualquer seqüência.

Quadro 3 - Códigos de evidência usados no genoma de *P. aeruginosa* (STOVER et al., 2000).

Al-Lazikani, Sheinerman e Honig (2001) usaram o termo "*certain"* para caracterizar domínios de estruturas *Janus Kinases* (JAK-SH2) que fossem correspondes e conhecidas no domínio da estrutura SH2.

Kasukawa et al. (2003), ao projetar o sistema de anotação FANTOM2 (*Functional Annotation of Mouse*), definiu 19 categorias identificadas numericamente para qualificar os genes seqüenciados, descritas no Quadro 4.

Qualifier	**Significado**	**Próximo estágio**
1	MGI (*Mouse Genome Informatics*) determinadas, indicando que a anotação foi concluída.	14 a 17
2	Similaridade maior ou igual a 98% considerando um tamanho maior ou igual a 100pb (pares de bases) com região codificante na área correspondente. Estas seqüências poderiam, após análise mais detalhada, receber as categorias de 14 até 17.	14 a 17
3	Similaridade maior ou igual a 98% considerando um tamanho maior ou igual a 100pb (pares de bases) sem uma região codificante na seqüência comparada. As categorias poderiam mudar para as categorias de 4 até 10 baseado somente em um porcentual de identidade e de uma fração referente ao tamanho das proteínas correspondentes.	4 a 10
4	98% ou mais de identidade em 100% do tamanho do gene.	11, 12 ou 13
5	85% ou mais de identidade em 100% do tamanho do gene.	11, 12 ou 13
6	85% ou mais de identidade em 90% do tamanho do gene	11, 12 ou 13
7	70% ou mais de identidade em 100% do tamanho do gene.	11, 12 ou 13
8	70% ou mais de identidade em 70% do tamanho.	11, 12 ou 13
9	50% ou mais de identidade em 100% do tamanho do gene.	11, 12 ou 13
10	50% ou mais de identidade em 50% do tamanho do gene.	11, 12 ou 13
11	Função atribuída pelo UniGene e TIGR.	
12	Função atribuída pelo UniGene.	
13	Função atribuída pelo TIGR.	
14	*Motif*[8] encontrado na base de dados do Interpro.	
15	*Motif* encontrado na base de dados MDS.	
16	*Motif* encontrado na base de dados SCOP.	

[8] Elemento de estrutura ou padrão repetido em diferentes proteínas.

17	Proteína hipotética, sem *motif* localizados em qualquer base de dados.	
18	EST's (*Expressed Sequence Tag*) com seqüências com similaridade significante nos bancos públicos. *"Unknown ESTs".*	
19	Falha em qualquer etapa anterior da verificação da proteína. *"Unclassifiable".*	

Quadro 4 - Códigos de evidência usados no sistema Fantom2.

Kasukawa et al. (2003) padronizou também a forma de descrever a proteína. Para as proteínas das categorias quatro, cinco e seis foi acrescentada a palavra-chave *"homolog"* ao nome da proteína correspondente; para as categorias sete e oito o nome foi anotado com o prefixo *"similar to"* ligando ao nome da seqüência encontrada e; para as categorias nove e dez foi usado o prefixo *"weakly similar to"*, indicando fraca similaridade com a proteína encontrada.

Searle et al. (2004) ao criar o sistema de anotação "*Otter*" estruturou sua base de dados de forma a permitir registro e suporte à evidência das anotações de cada gene. A evidência para cada transcrição foi representada por um nome e um tipo, não sendo relatado quais seriam estes tipos. Na conclusão de seu próprio trabalho relatou: "o atual nível de transcrição para suportar evidências não é ideal para determinar exatamente porque uma anotação em particular foi criada".

Categoria	**Evidência**
Valid (Y)	Atribuído a um produto genético bem definido, com elevada similaridade e homologia com ORF também válida ou com função reconhecida, independente do organismo.
Conserved Hypothetical (C)	Atribuído aos genes que tivessem elevada similaridade com outra ORF conservada ou pouca similaridade com uma ORF válida em outro organismo.
Hypothetical (U)	Atribuído aos genes que tivessem encontrado resultados não significantes retornados pelo programa BLAST, sem similaridade significante.
Invalid (N)	Atribuído às ORFs com sobreposição menor que 10 aminoácidos de outro organismo ou tamanho menor que 50 aminoácidos, quando não fosse considerado um gene.

Quadro 5 - Códigos de evidência usados no genoma de *C. violaceum.*

Almeida et al. (2004a) ao relatar características do projeto SABIÁ, desenvolvido primeiramente para anotação da bactéria *Chromobacterium violaceum*, adotou quatro categorias usadas no Projeto Genoma Brasileiro para classificar ORF's (*Open Read Frames*) ou genes. Os anotadores escolhiam uma categoria ao avaliar as informações automaticamente anotadas, e preenchiam o campo

“validation” da página de anotação com os códigos de evidência apresentados no Quadro 5.

Berardini et al. (2004) ao descrever a anotação funcional do genoma de *Arabidopsis* fez uso do vocabulário controlado definidos pelo GO e de 11 dos 14 códigos de evidência também definidos pelo GO, acrescentando ainda duas informações: “descrição da evidência” e “referência”, afirmando que a combinação destas informações “definem a base para a anotação e provê a informação necessária para um usuário interpretar uma anotação corretamente” e que “a descrição provê informação adicional à evidência usada para dar suporte à anotação”. As “descrições das evidências” fazem parte de um vocabulário controlado composto por 107 descrições. As anotações não revisadas por um curador foram assinaladas com o código “IEA” e as outras receberam o código conforme o experimento usado para fazer a associação com a evidência.

Rey et al. (2004) ao seqüenciar o genoma da bactéria *Bacillus licheniformis,* usou, ao exibir as seqüências correspondentes dos genes anotados, as categorias termos *“conserved hypothetical”* e *“hypothetical genes”* no campo de anotação, como em outros projetos genoma descritos anteriormente neste trabalho.

Liu, Hu e Wu (2005) desenvolveram a ferramenta de busca e visualização DynGO e, nesta ferramenta, criaram opção para que os usuários pudessem filtrar, organizar e visualizar as informações usando os códigos de evidência definidos pelo GO como critério.

Saxonov, Berg e Brutlag (2006), ao analisarem o genoma humano consideraram em seu trabalho somente anotações experimentalmente confirmadas, cujos códigos de evidência fossem correspondentes a IDA, IEP, IGI, IMP, IPI, ISS, e TAS do GO.

Aubry et al. (2006) ao propor um método para anotação funcional de genes combinando evidência, literatura biomédica e estatística usou os códigos de evidência do GO para fornecer resultados estatísticos de seu trabalho, sendo que 38,5% dos genes foram caracterizados por inferência eletrônica e foram associados ao código de evidência IEA. O restante, 61,5% dos genes, foi revisado por curadores e foram distribuídos em 39% para código de evidência TAS, 10,8% para NAS, 7,3% para IC, IDA, IEP, IGI, IMP, IPI e ISS; ainda 3,4% para NR e 1% para ND.

Xiang, Zheng e He (2006) desenvolveram o *Brucella Bioinformatics Portal* (BBP), composto por vários programas com função de integrar os dados genéticos

de *Brucella* e ferramentas de análise com literatura e um sistema de validação das informações. O BBP organizava as informações extraídas da literatura e de bases de dados públicas e permitia aos pesquisadores buscar, analisar e validar os dados do genoma de *Brucella*. Especialistas verificavam os dados manualmente para confirmar se uma possível interação entre os dados era possível e marcavam os dados com um código de evidência do GO indicando a evidência encontrada.

Bryson et al. (2006) ao implementar o sistema de anotação para bactérias denominado AGMIAL, usou qualificadores para controlar os dados gerados de forma automática e manual. O sistema AGMIAL é composto por dois subsistemas: *Contig Analysis Manager* (CAM) *e Protein Analysis Manager* (PAM). O qualificador é configurado no campo *status* do sistema e pode receber os valores: (1) *"original",* usado quando uma proteína era transferida do subsistema CAM para o subsistema PAM; (2) *"automatic",* definido após o subsistema PAM analisar proteína e sugerir automaticamente uma função a ela e; (3)*"confirmed",* definido por um curador após verificar os dados disponibilizados e o resultado gerado automaticamente, qual poderia alterar o *status* de *"automatic"* para *"confirmed*". Bryson et al. (2006) relatou também dados incorretos sobre a plataforma de anotação SABIÁ quando comparou suas características com as de outras plataformas de anotação, pois, ao contrário do relatado (1) a plataforma de anotação SABIÁ possui interface gráfica; (2) permite que os dados sejam validados por curadores, permite anotação manual; e (3) é um sistema que pode ser executado de forma colaborativa entre diversos pesquisadores.

Kulikova et al. (2006), relatando uma das mais importante base de dados do mundo, o EMBL, destaca algumas alterações recentes "para permitir ao usuário ver as evidências para uma anotação em particular e fazer um sábio julgamento sobre sua validade", objetivo este semelhante aos proposto neste trabalho. As alterações consistem em remover do projeto o qualificador *"evidence"* e criar dois outros qualificadores no lugar deste: (1) o qualificador *"experiment"* que é um campo de livre contexto onde deve ser relatada a técnica experimental usada e, (2) *"inference"* que é um qualificador altamente estruturado que detalha como a anotação foi inferida. Uma lista completa da estrutura e valores válidos para o qualificador "*inference"* está disponível no Instituto Europeu de Bioinformática no link http://www.ebi.ac.uk/embl/WebFeat/index.html (*European Bioinformatics Institute*, 2008).

A plataforma FlyBase (Flybase - **Reference Manual G,** 2008), na qual foi anotado o genoma de *Drosophila melanogaster,* faz uso dos códigos de evidência em sua documentação, relatando que é preferível associar os códigos baseados em evidência experimental (IMP, IGI, IDA, IPI, IEP), sendo que IEP é usado com baixa freqüência. Os códigos de evidência baseados em predição computacional (ISS, IEA, RCA), declaração de autor (NAS, TAS) e inferido por curador (IC) continuam sendo usados na ausência de dados experimentais, porém com a pretensão de serem removidos no futuro.

3 METODOLOGIA

Para determinar os códigos de evidência em anotação gênica e validar este trabalho, foi utilizado o genoma da bactéria *C. violaceum*, o qual se encontra anotado e validado por curadores (ALMEIDA et al., 2004a). A base de dados necessária à realização deste trabalho foi cedida pelo LNCC e foi instalada em um computador servidor na PUCPR, campus de Curitiba.

Para explicar como os objetivos propostos neste trabalho foram atingidos a metodologia foi dividida em três partes: (1) desenvolvimento do software GO-*based software for assign evidence codes of annotated genes* (GO-SIEVe); (2) atribuição dos códigos de evidência no genoma de *C. violaceum* usando o GO-SIEVe; e (3) a validação do processamento feito pelo GO-SIEVe.

3.1 DESENVOLVIMENTO DO GO-SIEVe

Para o desenvolvimento do GO-SIEVe foram utilizados recursos de hardware, uma base de dados e ferramentas de desenvolvimento, os quais estão descritos nas próximas seções:

3.1.1 Hardware

Para a elaboração deste trabalho foi utilizado um conjunto de hardware detalhado a seguir, entretanto, qualquer *hardware* compatível com os *softwares* poderia ser utilizado:

- cliente *notebook* Toshiba Satellite com sistema operacional *windows XP Professional*™, no qual o código fonte foi desenvolvido e testado;
- estação de trabalho no Laboratório de Informática em Saúde (LAIS) na PUCPR com sistema operacional *windows*™, onde o programa foi executado;

- servidor Sun modelo Sun Fire V880 com sistema operacional Solaris, o qual teve a função de servidor de BD através da instalação do SGBD PostgreSQL.

3.1.2 Sistema Gerenciador de Banco de Dados

Como recurso para armazenamento dos dados foi utilizado o SGBD PostgreSQL versão 8.3.1, distribuído sob licença *open source,* acessível pela linguagem de programação Java por meio do *driver* de conexão JDBC (*Java Data Base Connectivity*) (Sun Microsystem, 2008).

Com versões disponíveis para sistemas operacionais Windows™, Linux e Solaris, o SGBD foi instalado no servidor Sun e acessado pelas estações de trabalho com uso da ferramenta PGAdmin III [9] (PostgreSQL, 2008).

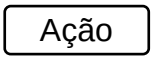
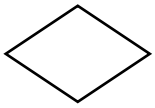
3.1.3 Softwares Usados para o Desenvolvimento de GO-SIEVe

Para o desenvolvimento do código fonte do programa GO-SIEVe foram utilizados os softwares listados abaixo, todos distribuídos sob licença gratuita:

- para o desenvolvimento do GO-SIEVe foi utilizado a linguagem de programação Java em sua versão 1.6.0, distribuída pela Sun Microsystems (2008);
- para fazer *download* dos arquivos da internet em formato *html* foi elaborada uma rotina com código específico no GO-SIEVe;
- as rotinas do GO-SIEVe com função de atribuir os códigos de evidência ao genes foram desenvolvidas diretamente na base de dados, usando a linguagem de programação plpgsql disponibilizada no SGBD Postgres (POSTGRESQL, 2008);
- como interface para o desenvolvimento da aplicação foi utilizado a ferramenta de desenvolvimento de software NetBeans IDE versão 6.0;

[9] Página para *download* do PGAdmin III http://www.pgadmin.org/download/

- para a modelagem dos dados foi utilizada a ferramenta *Java and UML Developers' Environment* (Jude)[10] em sua versão Community 3.2.1 compatível com *Unified Modeling Language* (UML) 2.0 (Object Management Group, 2008) e linguagem de programação Java;
- para modelar os aspectos dinâmicos do GO-SIEVe foi utilizado diagrama de atividade, o qual enfatiza o fluxo de controle de uma atividade do software para outra através das formas mostradas no Quadro 6 (BOOCH; RUMBAUGH; JACOBSON, 2000):

Símbolo	Significado
	Símbolo que marca o início das atividades.
	Símbolo que representa a transição de uma atividade para outra, mudança para próximo estado da atividade.
	Símbolo que representa uma atividade, usado para descrever a ação de uma atividade.
	Símbolo que representa um estado de decisão. Caminhos diferentes podem ser seguidos dependendo do resultado da decisão.
	Símbolo que marca o término das atividades.

Quadro 6 - Descrição da simbologia utilizada nos diagramas de atividades.

3.2 ATRIBUIÇÃO DOS CÓDIGOS DE EVIDÊNCIA

Todo o processo de atribuição dos códigos de evidência na anotação de *C. violaceum* foi executado pelo programa GO-SIEVe, sem intervenção humana, sendo que somente os genes válidos (com *status* “C”, “V” e “U” no SABIÁ) anotados e revisados foram computados.

O processo de atribuição dos códigos de evidência pelo GO-SIEVe foi executado em quatro etapas: (1) preparar a base de dados; (2) recuperar os *links*

[10] Página para download: http://jude.change-vision.com/jude-web/index.html

nas páginas do BLAST referente a cada gene válido armazenado na base de dados do SABIÁ (3) recuperar os “títulos” e “resumos” no NCBI e; (4) atribuir os códigos de evidência conforme os termos fornecidos pelos especialistas.

O processo geral de atribuição dos códigos de evidência está representado pelo diagrama de atividades da Figura 4 e, a descrição do fluxo de atividades que o GO-SIEVe faz está representado pelo fluxograma da Figura 5, descritos com mais detalhes nas próximas seções.

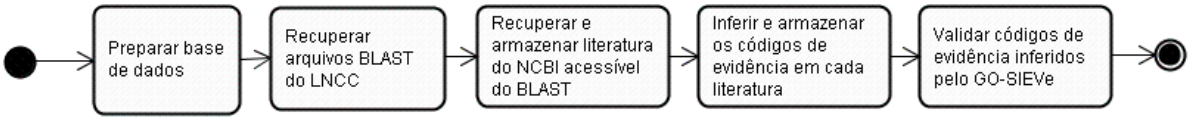


Figura 4 - Diagrama de atividades: Etapas executadas pelo GO-SIEVe para atribuir códigos de evidência.

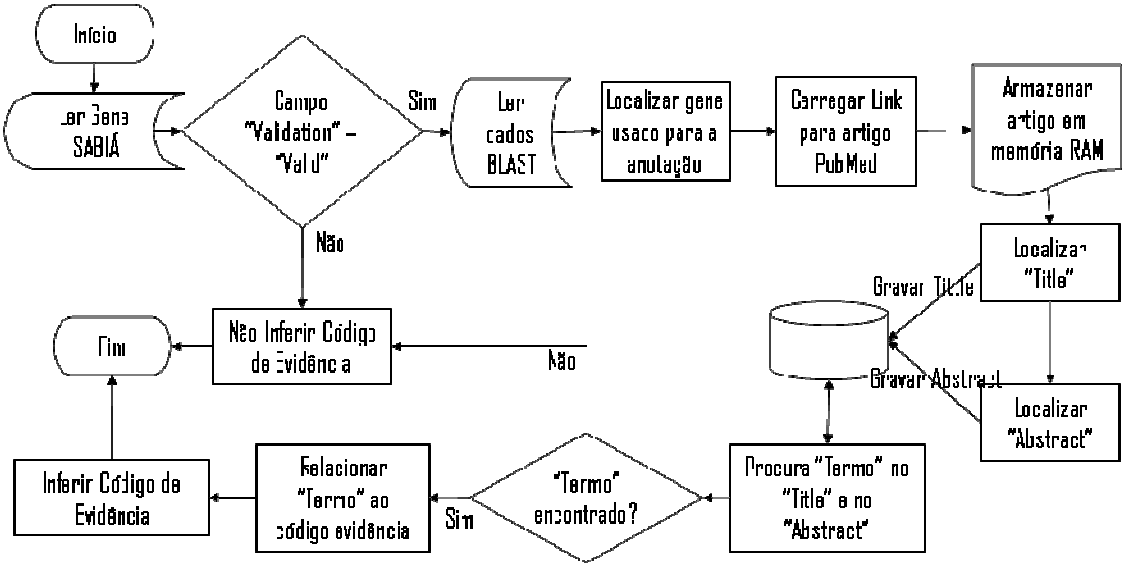


Figura 5 - Fluxograma - Processo de atribuição do código de evidência.

3.2.1 Preparação da Base de Dados

A base de dados do projeto SABIÁ foi cedida pelo LNCC e instalada sob o SGBD PostgreSQL. Para que as informações pudessem ser armazenadas, processadas e posteriormente recuperadas foi necessário definir a estrutura da base de dados através da criação de tabelas (relações) de dados (SILBERSCHATZ; KORTH; SUDARSHAN, 2006).

Foram adicionadas cinco tabelas aos dados da plataforma SABIÁ e todas foram geradas conforme *scripts* apresentados no APÊNDICE H. Cada tabela exerce uma finalidade específica, e são:

- tabela InfoBlast, usada para armazenar as informações recuperadas das páginas do BLAST;
- tabela InfoNcbi, usada para armazenar o endereço, os títulos e resumos dos artigos recuperados do NCBI;
- tabela EvidenceLevel, usada para armazenar os códigos de evidência definidos pelos especialistas, os quais serão atribuídos no genoma de *C. violaceum*;
- tabela EvidenceTermo, usada para armazenar os termos e sua relação com cada código de evidência, os quais foram buscados nos títulos e resumos dos artigos recuperados do NCBI para atribuição de um código de evidência;
- tabela EvidenceRegra, usada para armazenar as regras de busca adicionais de cada termo.

O conteúdo das tabelas EvidenceLevel, EvidenceTermo e Evidence Regra foram definidos pelos especialistas e formam a base para a busca e atribuição dos códigos de evidência propostos pelo GO-SIEVe. Já as tabelas InfoBlast e InfoNcbi tiveram seu conteúdo definido pelo processamento feito pelo GO-SIEVe sobre os genes de *C. violaceum*.

3.2.2 Recuperar *Links* do BLAST para o NCBI

Os arquivos *html* resultantes do processamento do programa BLAST estão armazenadas em um servidor no LNCC e são acessíveis via *internet*[11]. Os arquivos referentes à anotação de cada gene válido de *C. violaceum* foram acessados e computados localmente pelo GO-SIEVe. Os arquivos do BLAST são os mesmos gerados durante o período de anotação de *C. violaceum* em 2003 e não são atuais.

Em cada arquivo do BLAST existem vários *links* que remetem aos artigos depositados no NCBI que contêm informações sobre cada gene similar encontrado pelo BLAST (Figura 3b). Os *links* de cada gene similar do BLAST foram recuperados

[11] Exemplo de um endereço para um arquivo BLAST armazenado no LNCC: http://www.brgene.lncc.br/webbie/annotation/BlastP_Final/68.blastp.html

e armazenados na tabela “InfoBlast” junto com o valor e-value e o valor de cobertura correspondente a cada gene (Figura 3c).

As atividades executadas pelo GO-SIEVe nos arquivos do BLAST e do NCBI estão representadas pelo diagrama de atividades da Figura 6.

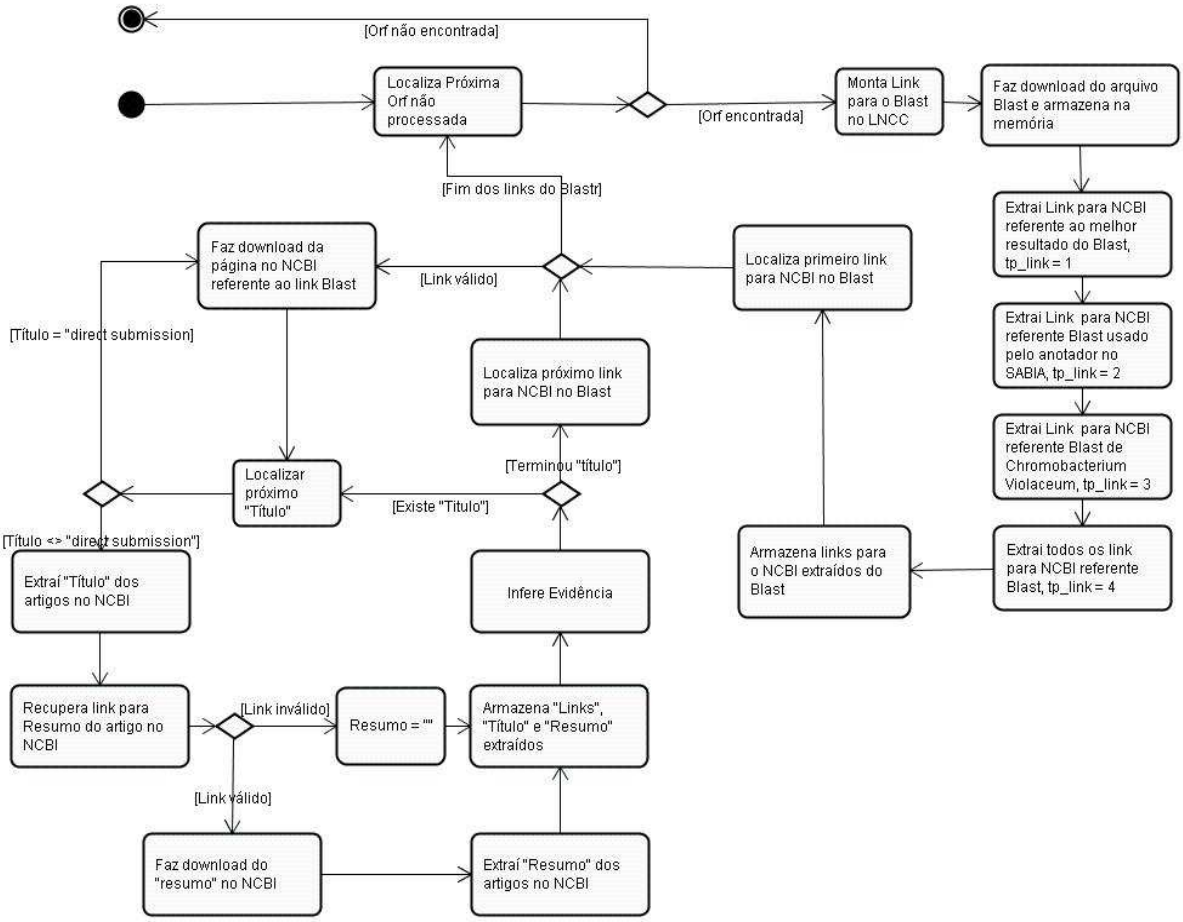


Figura 6 - Diagrama de atividades: processamento do BLAST e artigos no NCBI.

A atividade “infere evidência” da Figura 6 está detalhada na Figura 7.

Como cada *link* do BLAST remete a arquivos em formato *html* referente a publicações sobre o gene no NCBI, é possível acessar estes arquivos no NCBI e recuperar dados, como títulos e resumos.

3.2.3 Recuperar dados das Páginas do NCBI

Em posse dos endereços recuperados dos arquivos BLAST para as páginas no NCBI[12], foi feito o *download* destas páginas e recuperadas as informações nelas contidas. Duas informações principais nestas páginas foram recuperadas: (1) “Título do artigo” (Figura 3d) identificado pela palavra “*TITLE*” e (2) o *link* para o “texto do resumo” armazenado no PubMed[13] (Figura 3e) identificado pela palavra “PUBMED”. Tanto o “título” do artigo quanto o “resumo”, doravante denominado simplesmente de “literatura”, foram extraídos de cada página e armazenados na tabela “InfoNcbi” para posterior processamento e atribuição dos níveis de evidência pelo GO-SIEVe.

Em cada página no NCBI pode existir de zero a várias “literaturas”, não havendo uma quantidade específica de informações recuperadas. Como fator de exclusão, os títulos denominados “*direct submission”* foram desconsiderados por serem artigos de submissão e não serem artigos que possam caracterizar os genes.

Estando a “literatura” referente aos artigos do NCBI armazenada em base de dados local, foi buscado nela informações para que um código de evidência fosse atribuído.

3.2.4 Atribuir Códigos de Evidência

Cada código de evidência pode ter relacionado a ele vários termos que o caracterizam, sendo que tanto os códigos de evidência quanto os termos foram definidos por especialistas.

Os códigos de evidência estão armazenados na tabela “EvidenceLevel” e os termos estão armazenados na tabela “EvidenceTermo”, sendo que a qualquer momento, novos códigos de evidência e termos podem ser alterados ou inseridos na base de dados acessada pelo GO-SIEVe. O Quadro 7 contém a relação de todos os códigos de evidência utilizados pelo GO-SIEVe, bem como os termos associados a cada código.

---

[12] Exemplo de um endereço para literatura armazenada no NCBI acessível por um *link* do BLAST: http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve &db=Protein&list_uids=16123858&dopt=GenPept

[13] Exemplo de um endereço para o resumo de um artigo no PubMed acessível por um *link* do NCBI: http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=pubmed&list_uids=12834539

Os termos definidos pelos especialistas foram cruzados com a literatura recuperada no NCBI e, ao ser encontrado um termo, o código de evidência relacionado a este termo foi atribuído ao gene.

EVIDÊNCIA	TERMOS	EVIDÊNCIA	TERMOS
CV=	violaceum	IGI	synthetic lethal
IDA	immunofluorescence	IGI	functional complementation
IDA	binding assay	IGI	rescue experiment
IDA	enzymatic activity	IMP	knockout experiment
IDA	binding experiment	IMP	knock-out experiment
IDA	direct assay	IMP	knockout assay
IDA	expressed in vivo	IMP	knock-out assay
IDA	deletion of the c-terminus	IMP	gain of function
IDA	deletion of the n-terminus	IMP	loss of function
IDA	kinetic analysis	IMP	deletion mutant
IDA	2-hybrid interaction	IMP	mutant phenotype
IDA	two-hybrid interaction	ISS	genome sequence
IDA	co-immunoprecipitation	ISS	genomic sequence
IDA	enzyme assay	ISS	the genome of
IDA	immunolocalization	ISS	gene product
IEP	expression profiling	ISS	sequence similarity
IEP	pattern of expression	ISS	similarity at sequence level
IEP	transcript level	ISS	protein structure similarity
IEP	level os transcript	ISS	structural similarity
IEP	expression pattern	ISS	total protein
IGC	operon organization	ISS	protein species
IGC	deduced from synteny	ISS	proteome
IGC	genomic context	ISS	laser desorption ionization
IGC	operon structure	ISS	maldi-tof
IGI	rescue assay	ISS	maldi-tof-ms
IGI	supressor experiment	ISS	deduced from aminoacid sequence
IGI	supressor assay	ISS	chromosome sequence

Quadro 7 - Códigos de evidência com os seus termos relacionados

As atividades executadas pelo GO-SIEVe para atribuir os códigos de evidência estão representadas pelos diagramas de atividades da Figura 7 e da Figura 8.

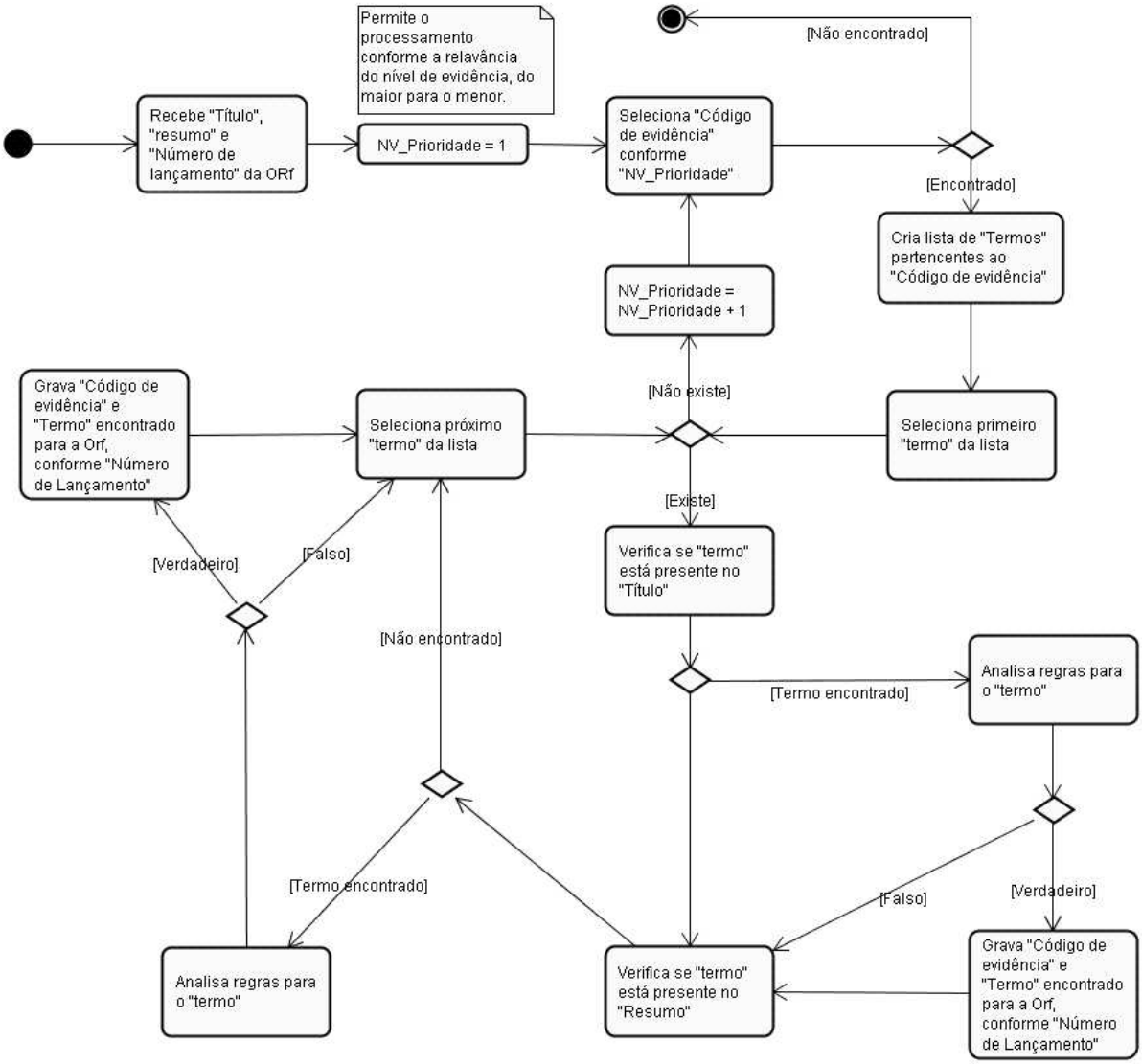


Figura 7 - Diagrama de atividades: infere evidência, sub processo do diagrama da Figura 6.

A atividade “analisa regras” do diagrama de atividades da Figura 7 está representada detalhadamente pelo diagrama de atividades da Figura 8.

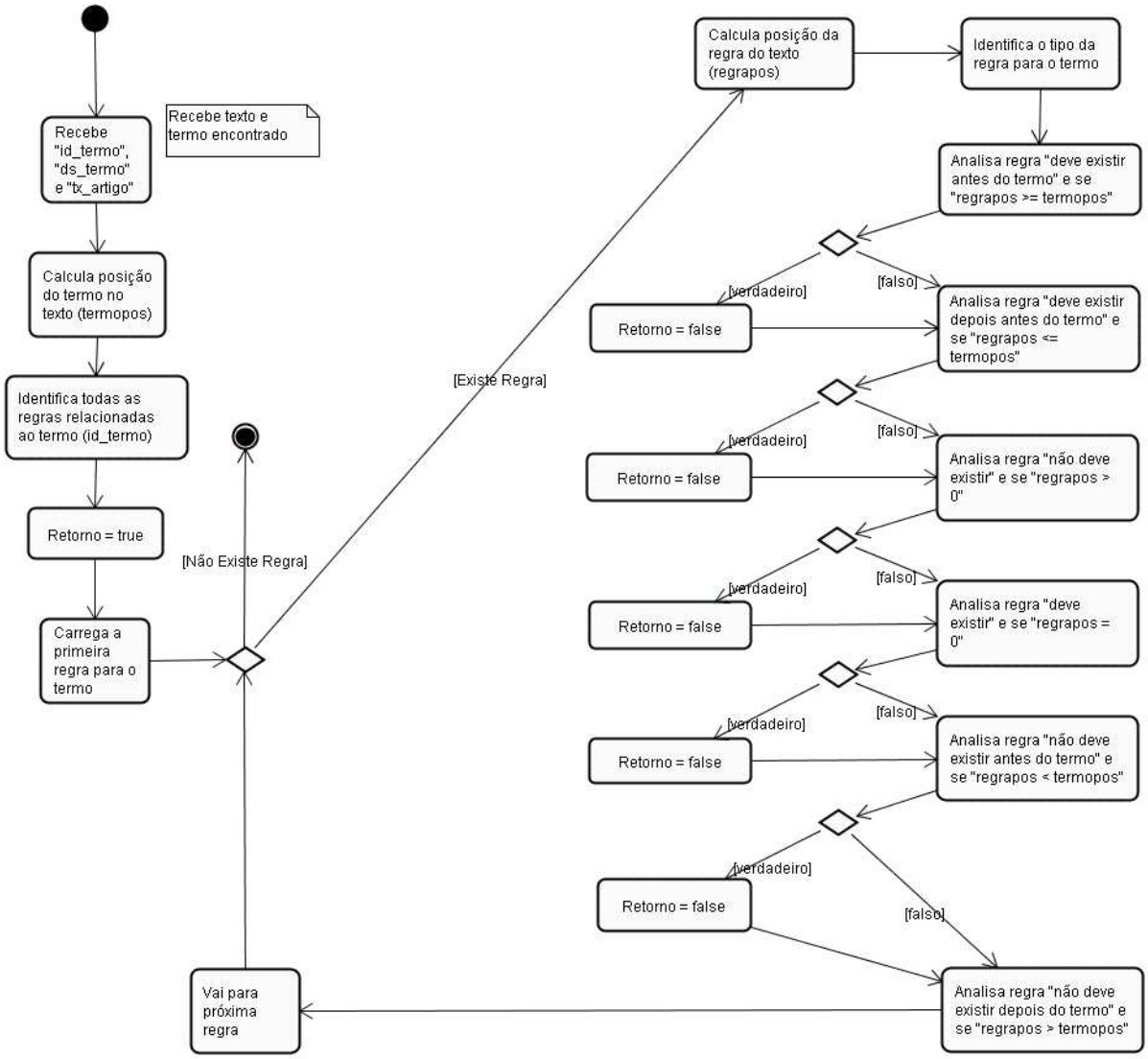


Figura 8 - Diagrama de atividades: analisa regras, sub processo do diagrama da Figura 7.

3.3 FATOR DE EXCLUSÃO

Foram utilizados para a execução deste trabalho os genes de *C. violaceum* anotados e revisados no SABIÁ considerados válidos, os quais possuem para o campo “*validation*” os valores “Y”, “C” e “U”, significando genes válidos, conservados e hipotéticos, respectivamente.

Foram excluídos deste trabalho os artigos encontrados no NCBI que tinham o título denominado “*direct submission”*, por ser depósito da seqüência de um organismo diretamente no NCBI, sem artigo de literatura correspondente.

3.4 VALIDAÇÃO

A validação do processamento feito pelo GO-SIEVe no genoma de *C. violaceum* foi feito por dois especialistas, os quais, pela leitura dos resumos associados a cada códigos de evidência, verificaram se os códigos de evidência atribuídos automaticamente pelo GO-SIEVe eram coerentes com o conteúdo apresentado na literatura.

Por processo de amostragem, foi validado o total de 74,4% das atribuições por cada especialista, sendo que cada um, conforme interpretação particular da literatura e do termo encontrado pelo GO-SIEVe, validou como verdadeira ou falsa a atribuição do código de evidência feito pelo GO-SIEVe, sendo o resultado final da validação a média de erros e acertos apontados pelos especialistas.

4 RESULTADOS

A descrição dos resultados deste trabalho foi dividida em duas partes. A primeira apresentando o GO-SIEVe como uma ferramenta e, a segunda, apresentando os resultados obtidos com o processamento do GO-SIEVe sobre o genoma de *C. violaceum*.

4.1 A FERRAMENTA GO-SIEVe

O software GO-SIEVe foi desenvolvido em dois módulos, sendo que nenhum é módulo gráfico, pois o processamento é feito com uso direto da internet e de banco de dados, sem a necessidade de interagir com o usuário. Cada módulo possui funções distintas:

- o primeiro módulo tem função de acessar, recuperar e armazenar a “literatura” encontrada no NCBI;
- o segundo módulo, desenvolvido diretamente na base de dados, tem a função de procurar os termos na literatura, armazenada pelo primeiro módulo e, atribuir um código de evidência correspondente ao termo encontrado.

Mesmo sendo possível desenvolver os dois módulos em um único módulo, optou-se em desenvolvê-los separadamente para deixá-los independentes e o software mais dinâmico, possibilitando a execução de cada módulo em momentos distintos. Assim, os códigos de evidência podem ser atribuídos após cada artigo ser armazenado, ou então, ser atribuído após todos os artigos serem armazenados; bem como, após todos os artigos serem armazenados, os códigos de evidência podem ser atribuídos quantas vezes for necessário. A última opção foi escolhida para permitir a melhoria constante dos termos conforme o resultado do processamento fosse validado, possibilitando aos especialistas eliminarem ou criarem novos “termos” e computarem novamente os artigos, obtendo novos resultados.

Ainda, para que o GO-SIEVe atingisse os objetivos propostos neste trabalho sete classes foram criadas e implementadas conforme apresentado na Figura 9, cada qual com uma função específica:

- classe Mensagem, responsável por interagir com o usuário mostrando o que o GO-SIEVe está executando e as mensagens de erros;
- classe Excecao, responsável por fazer o tratamento de erros do GO-SIEVe e emitir alerta ao usuário;
- classe Conexão, responsável por fazer e manter a conexão com a base de dados do sistema, bem como executar os comando de leitura e gravação emitidos na linguagem SQL;
- classe ProcessaArquivo, responsável por receber o endereço de uma página de internet (*link*) e fazer o *download*, carregando-a na memória do computador e disponibilizando para ser computada. É a classe responsável por acessar os arquivos do BLAST no LNCC e páginas que contém a "literatura" no NCBI;
- classe SabiatoBlast, responsável por selecionar os genes válidos ainda não computados na base de dados do SABIÁ e montar o endereço (*link*) para o arquivo BLAST armazenado no LNCC correspondente à anotação;
- classe ProcessaBlast, responsável por analisar cada arquivo BLAST recuperado, acessar, montar e armazenar o endereço (*link*) para a literatura armazenada no NCBI referente a cada gene similar encontrado;
- classe ProcessaNcbi, responsável por analisar cada arquivo referente aos *links* do BLAST recuperados no NCBI, acessar e armazenar a "literatura" encontrada.

O relacionamento entre todas as classes, bem como seus atributos e métodos, são mostrados no diagrama de classes (BOOCH; RUMBAUGH; JACOBSON, 2000) da Figura 9 e, o código fonte de todas as classes que compõem o GO-SIEVe está disponível no APÊNDICE I deste trabalho.

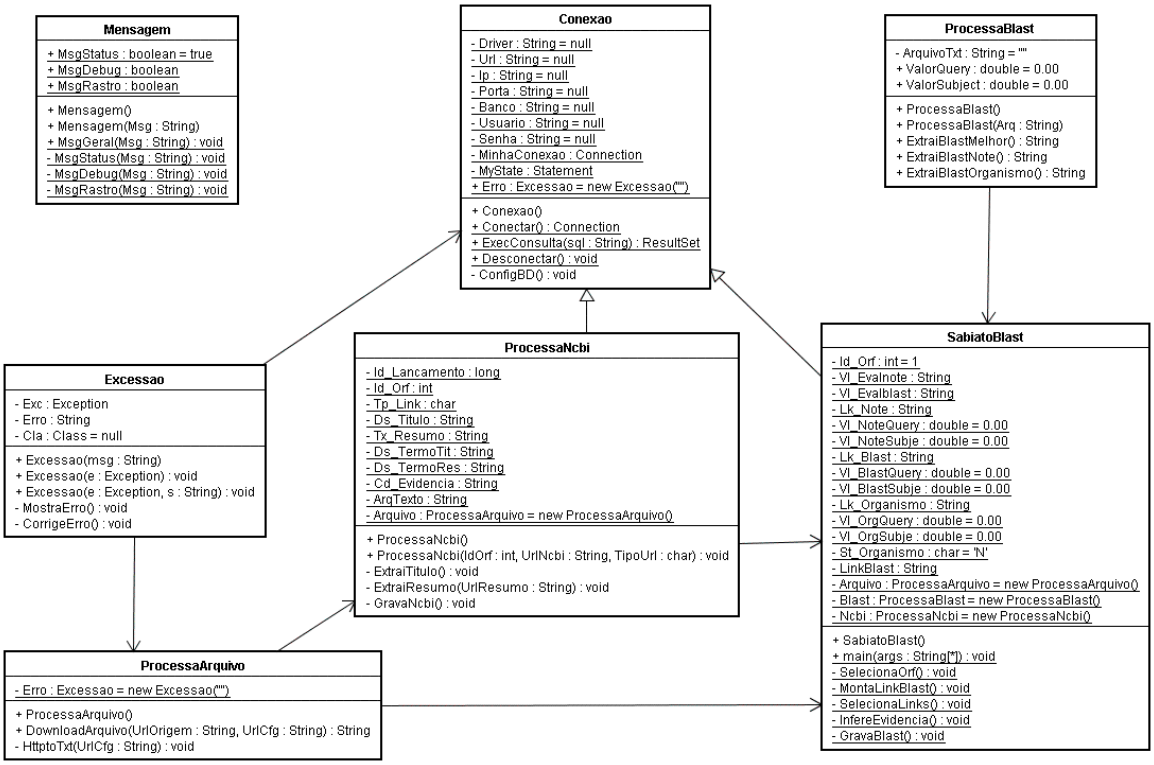


Figura 9 - Diagrama de Classes do GO-SIEVe.

4.2 CÓDIGOS DE EVIDÊNCIA ATRIBUIDOS PELO GO-SIEVe

A execução do GO-SIEVe foi iniciada no dia dezoito de outubro de 2007 e o processamento terminou no dia oito de fevereiro de 2008. Em 49 dias descontínuos de execução recuperou e armazenou em média 3.696 “literaturas” por dia, conforme Quadro 15 no APÊNDICE A deste trabalho.

O genoma de *C. violaceum* anotado no SABIÁ possui 4.431 genes validados por curadores e categorizados como: 2.717 genes válidos, 958 genes conservados e 756 genes hipotéticos (VASCONCELOS et al., 2003). Para estes 4.431 genes processados pelo GO-SIEVe, foram recuperadas e armazenadas na base de dados 181.098 “literaturas” (títulos e/ou resumos de artigos), sendo que deste total 105.204 (58,1%) “literaturas” tiveram algum código de evidência atribuído e 75.894 (41,9%) não tiveram qualquer código de evidência atribuído. O Gráfico 1 mostra a quantidade de genes e de literatura recuperada para cada categoria validada.

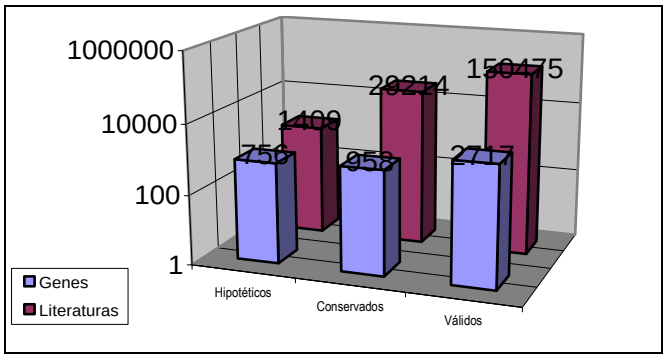


Gráfico 1 - Quantidade de literaturas baixadas por categorias.

4.2.1 Resultados Considerando os Códigos de Evidências

A literatura para a qual foi encontrado algum termo e foi atribuído algum código de evidência está quantificada e distribuída pelos códigos de evidência conforme apresentado no Gráfico 2. Para o total de 105.204 (58,1%) literaturas onde algum termo foi encontrado pelo GO-SIEVe, foram atribuídos 121.115 códigos de evidência.

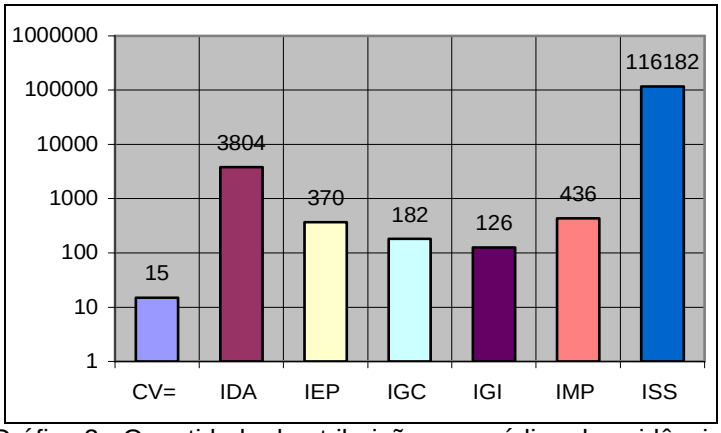


Gráfico 2 - Quantidade de atribuições por código de evidência.

Considerando que foram atribuídos mais que um código de evidência para algumas literaturas (dois, três ou até quatro códigos de evidência), a quantidade de códigos de evidência atribuídos pelo GO-SIEVe foi maior que a quantidade de literaturas armazenadas. Como algumas literaturas apresentaram mais de um código de evidência, foram gerados agrupamentos de códigos de evidência, sendo que, a soma destes agrupamentos, confere com o total de “literatura” na base de dados, como pode ser verificado no Quadro 16 do APÊNDICE B deste trabalho.

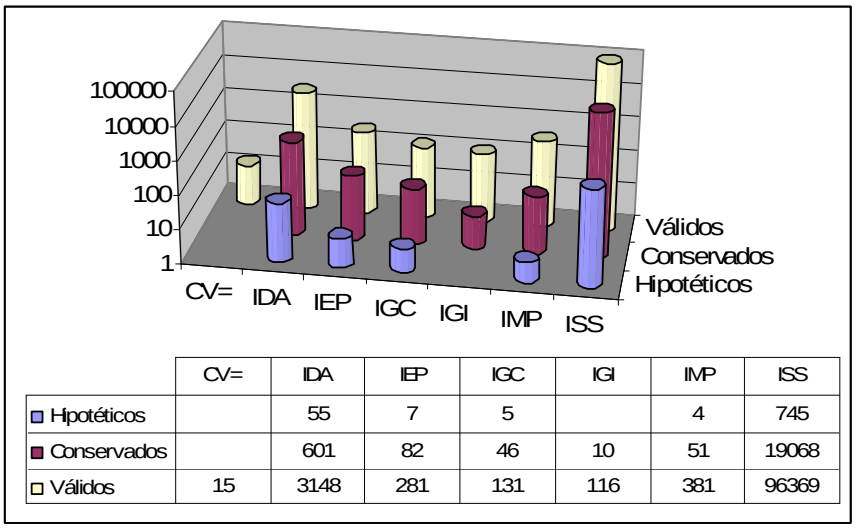
Como exemplo desta repetição de códigos de evidência na literatura, é possível observar que no Gráfico 2 existe 126 literaturas para o código de evidência IGI. No entanto, para estas 126 literaturas, os códigos de evidências ficaram distribuídos conforme o Quadro 16 (APÊNDICE B), somando-se para IGI:

- 32 atribuições para o agrupamento de evidência IGI e ISS;
- 80 atribuições para o agrupamento de evidência IGI;
- 1 atribuição para o agrupamento de evidência IGI e IGI;
- 1 atribuição para o agrupamento de evidência IDA, IGI e ISS;
- 4 atribuições para o agrupamento de evidência IDA, IGC, IGI e ISS;
- 1 atribuição para o agrupamento de evidência IGI, IMP e ISS;
- 2 atribuições para o agrupamento de evidência IDA e IGI;
- 3 atribuições para o agrupamento de evidência IGI e IMP; e
- 2 atribuições para o agrupamento de evidência IEP e IGI.

As 126 literaturas do Quadro 16 estão computadas para outros códigos de evidência também. Estes agrupamentos de evidências, destacados no Quadro 16 (APÊNDICE B), ocorreram devido a duas situações:

- um termo foi encontrado no “título” e outro termo no “resumo” da mesma “literatura”, sendo atribuído código de evidência a ambos ou;
- foi encontrado no “título” ou no “resumo” vários (mais de um) termos relacionados a diferentes códigos de evidências. Apesar de mais raro, foi encontrado “literatura” onde foi identificado até três termos em seu conteúdo, seja no “título”, “resumo” ou em ambos.

O genoma de *C. violaceum* foi validado por curadores, os quais categorizaram os genes, sendo que a distribuição dos códigos de evidência para cada categorias é apresentada no Gráfico 3.



	CV=	IDA	IEP	IGC	IGI	IMP	ISS
Hipotéticos		55	7	5		4	745
Conservados		601	82	46	10	51	19068
Válidos	15	3148	281	131	116	381	96369

Gráfico 3 - Distribuição das evidências atribuídas pelo GO-SIEVe pelas categorias dos genes.

4.2.2 Resultados considerando os Termos

Os resultados foram analisados também considerando cada termo cadastrado, fornecendo detalhes que caracterizam as evidências encontradas. Foram cadastrados no GO-SIEVe 54 termos, distribuídos em sete códigos de evidência, conforme apresenta o Quadro 7.

Dos 54 termos cadastrados na base de dados do GO-SIEVe, 39 foram encontrados na literatura, conforme pode ser observado no APÊNDICE C deste trabalho. O Quadro 8 mostra a quantidade de vezes que cada termo foi encontrado na literatura (atribuições), relacionado com as categorias para as quais o gene foi validado no SABIÁ.

Termo	Conservado	Hipotético	Válido	Total
binding assay	24	3	63	90
binding experiment	2		6	8
chromosome sequence	379	15	1818	2212
co-immunoprecipitation	1		5	6
deletion mutant	34	3	272	309
direct assay	1		4	5
enzymatic activity	30	4	202	236
enzyme assay	6		70	76
expressed in vivo	367	19	1840	2226
expression pattern	28	5	125	158
expression profiling	47	2	78	127
functional complementation	7		103	110
gain of function			3	3
gene product	591	22	3984	4597
genome sequence	11285	413	57134	68832
genomic context	3	1	3	7
genomic sequence	2283	129	10693	13105
immunofluorescence	15	12	74	101
immunolocalization	2	1	8	11
kinetic analysis	153	16	876	1045
knockout experiment			4	4
knock-out experiment	1		1	2
laser desorption ionization	4	1	5	10
loss of function	3		25	28
maldi-tof	4		11	15
mutant phenotype	13	1	76	90
operon organization	40	3	106	149
operon structure	3	1	22	26
pattern of expression	3		31	34
protein species	1		7	8
proteome	423	26	2337	2786
rescue experiment	3		5	8
sequence similarity	401	10	2415	2826
structural similarity	4		46	50
synthetic lethal			8	8
the genome of	3484	124	17050	20658
total protein	209	5	869	1083
transcript level	4		47	51
violaceum			15	15
Total geral	**19858**	**816**	**100441**	**121115**

Quadro 8 - Quantidade de evidências por termos cadastrados e categoria dos genes.

Os termos cadastrados foram procurados na literatura recuperada a partir do resultado BLAST de cada gene e, ao ser encontrado algum termo, um código de evidência foi atribuído ao gene que possuía em sua literatura o respectivo termo.

Evidências	Total
	24
:IDA:ISS	10
:IEP	1
:IMP	1
:ISS	13
:ISS:ISS	20
Total geral	**69**

Quadro 9 - Exemplo de códigos de evidência atribuídos ao gene válido CV2101, anotados na plataforma SABIÁ.

No Quadro 9 são exibidos os códigos de evidência atribuídos ao gene CV2101, categorizado como um gene válido, anotado na plataforma SABIÁ com

código seqüencial de anotação 1925. Somente para este gene foram recuperadas 69 literaturas, e nestas, foram encontrados 45 termos, possibilitando a atribuição dos códigos de evidência IDA, ISS, IEP e IMP a um único gene. Esta característica é predominante no genoma de *C. violaceum*. Detalhes destes 69 códigos de evidência atribuídos ao gene CV2101 podem ser verificados no Quadro 20 do APÊNDICE E.

No Quadro 10 é exibido apenas 3 das 69 literaturas recuperadas para o gene CV2101 e, para cada literatura (título e resumo) estão associados os termos encontrados e os códigos de evidência atribuídos ao gene. Iguais a estas três literaturas de amostra, o GO-SIEVe recuperou e armazenou na base de dados 181.098 literaturas para os 4.431 genes, sendo que deste total 105.204 (58,1%) literaturas tiveram algum código de evidência atribuído e 75.894 (41,9%) não tiveram qualquer código de evidência atribuído.

Exemplo de atribuição dos códigos de evidência com termos e literatura		
Código: 1925	**Categoria: válido**	**Gene:CV2101**

Evidências: IDA, ISS
Termos: the genome of, kinetic analysis
Título: Discovery, characterization and **kinetic analysis** of an alditol oxidase from streptomyces coelicolor
Resumo: A gene encoding an alditol oxidase was found in **the genome of** Streptomyces coelicolor A3(2). This newly identified oxidase, AldO, was expressed at extremely high levels in Escherichia coli when fused to maltose-binding protein. AldO is a soluble monomeric flavoprotein with subunits of 45.1 kDa, each containing a covalently bound FAD cofactor. From sequence alignments with other flavoprotein oxidases, it was found that AldO contains a conserved histidine (His(46)) that is typically involved in covalent FAD attachment. Covalent FAD binding is not observed in the H46A AldO mutant, confirming its role in covalent attachment of the flavin cofactor. Steady-state **kinetic analyses** revealed that wild-type AldO is active with several polyols. The alditols xylitol (K(m) = 0.32 mm, k(cat) = 13 s(-1)) and sorbitol (K(m) = 1.4 mm, k(cat) = 17 s(-1)) are the preferred substrates. From pre-steady-state **kinetic analyses**, using xylitol as substrate, it can be concluded that AldO mainly follows a ternary complex kinetic mechanism. Reduction of the flavin cofactor by xylitol occurs at a relatively high rate (99 s(-1)), after which a second kinetic event is observed, which is proposed to represent ring closure of the formed aldehyde product, yielding the hemiacetal of d-xylose. Reduced AldO readily reacts with molecular oxygen (1.7 x 10(5) m(-1) s(-1)), which confirms that the enzyme represents a true flavoprotein oxidase.<br

Evidências: IMP
Termos: deletion mutant
Título: Modulating factors for the Pkn4 kinase cascade in regulating 6-phosphofructokinase in Myxococcus xanthus
Resumo: Myxococcus xanthus, a Gram-negative developmental bacterium, contains a large number of protein Ser/Thr kinases (PSTKs). Among these PSTKs, Pkn4 has been shown to be 6-phosphofructokinase (PFK) kinase. PFK associates with the regulatory domain of Pkn4 (Pkn4RD) and is activated by Pkn4-mediated phosphorylation. The activation of PFK is required to consume glycogen accumulated during early development and is essential for efficient sporulation. Using the yeast two-hybrid screen, we identified three new factors, MkapA, MkapB and MkapC, that interact with Pkn4 and each contains well-known protein-protein interaction domains. MkapB contains eight tandem repeats of the TPR (tetratrico peptide repeat) domain and its interaction with Pkn4RD was phosphorylation-dependent. MkapB remained associated with Pkn4RD. As a result, Pkn4 did not interact with PFK and its activation was inhibited. While deletion of the pfk-pkn4 operon did not inhibit fruiting body formation, the spore yield was low. In contrast, a mkapB **deletion mutant**

exhibited a 24 h delay in fruiting body formation, accumulated less glycogen in the stationary phase and gave rise to 3.2% spore formation as opposed to 100% attained with DZF1. In addition to Pkn4, MkapA associated with other membrane-associated PSTKs, Pkn1, Pkn2, Pkn8 and Pkn9, while MkapB associated with Pkn8 and Pkn9, and MkapC with Pkn8. These results indicate that there are complex PSTK networks in M. xanthus that share common modulating factors.

Evidências: IEP
Termos: expression pattern
Título: Expression of the Streptomyces aureofaciens glyceraldehyde-3-phosphate dehydrogenase gene (gap) is developmentally regulated and induced by glucose
Resumo: In previous experiments, the Streptomyces aureofaciens gap gene encoding glyceraldehyde-3-phosphate dehydrogenase (GAPDH) was identified. To investigate expression of the gene, S1 nuclease mapping and Northern blot hybridization were performed using RNA prepared from S. aureofaciens cultivated under various conditions. These studies suggested monocistronic organization and developmental regulation of the gene. A single promoter, gap-P, was identified upstream of the gap coding region. In cultures grown on solid medium in the absence of glucose, its transcription was induced at the time of aerial mycelium formation. In addition, gap transcription was also induced in substrate mycelium by glucose. A promoter-bearing DNA fragment was inserted into two promoter-probe vectors, to give **expression pattern**s consistent with the results of direct RNA analysis.

Quadro 10 - Amostra dos códigos de evidência atribuídos e os termos relacionados a três literaturas encontradas para CV2101.

Das 121.115 atribuições na literatura recuperada pelo BLAST para os 4.431 genes de *C. violaceum* a maioria se repete entre os genes, ou seja, um artigo que apareceu para o gene X pode ter aparecido para o gene Y. Somente a soma das 60 literaturas mais repetidas no GO-SIEVe totalizam 90.034 literaturas, como pode ser observado no Quadro 18 do APÊNDICE D. O Quadro 11 mostra o total de atribuições na literatura, o total de literatura que não se repete e o percentual de repetição em cada código de evidência atribuído. O Quadro 19 do APÊNDICE D agrupa a literatura e suas repetições por cada código de evidência e seus termos, detalhando os valores apresentados no Quadro 11.

Neste trabalho, foi computado como “literatura recuperada”, todos os títulos e resumos armazenados na base de dados do GO-SIEVe e, em caso de alguns destes artigos aparecerem mais de uma vez (repetidos), foi contado como um artigo na “literatura não repetida”.

Total de Literatura e Repetições			
Código Evidencia	Literatura recuperada	Literatura não repetida	% Repetição
CV=	15	8	46,7
IDA	3804	323	91,5
IEP	370	164	55,7
IGC	182	23	87,4
IGI	126	54	57,1
IMP	436	158	63,8
ISS	116182	1787	98,5
Total geral	**121115**	**2517**	**97,9**

Quadro 11 - Total de literatura e suas repetições agrupadas por evidência.

4.2.3 Validação do Resultado

Do total de 2.517 atribuições na literatura sem repetição, 1.889 foram validadas pelos especialistas, totalizando 75,0% das atribuições e, deste montante validado, houve uma média de 82,1% de acerto pelo GO-SIEVe, mostrado no Quadro 12. O maior índice de acerto pelo GO-SIEVe foi para o código de evidência IMP, com 93,4% de acerto e, o menor índice de acerto foi para o código de evidência CV=, com 68,8% de acerto. Neste trabalho, o total validado corresponde a quantidade da literatura que foi validada pelos especialistas e, o percentual validado, indica o percentual da literatura armazenada que foi validada (% validado = total validado / literatura única armazenada * 100). Já o percentual de acerto indica quantas atribuições o GO-SIEVe acertou em relação ao que foi validado pelos especialistas. Por exemplo, para o código de evidência CV=, onde foram validadas oito atribuições, o percentual de acerto de 68,8% corresponde à média de acerto de 5,5 atribuições (8 * 68,8 / 100).

Resultado da validação				
Código de Evidencia	Literatura armazenada	Total validado	% validado	% de acerto
CV=	8	8	100,0	68,8
IDA	323	237	73,4	85,3
IEP	164	94	57,3	71,4
IGC	23	23	100,0	73,6
IGI	54	52	96,3	86,7
IMP	158	113	71,5	93,4
ISS	1787	1362	76,2	84,0
Total geral	**2517**	**1889**	**75,0**	**82,1**

Quadro 12 - Resultado da validação agrupado por código de evidência.

Uma visão detalhada do Quadro 12 é mostrada no Quadro 13, o qual agrupa os resultados pelos termos de cada código de evidência, mostrando o total de literatura existente, o total validado e a margem de acerto do GO-SIEVe de cada termo. O “% de acerto” do Quadro 12 e do Quadro 13 é o resultado da avaliação feita pelos especialistas considerando os códigos de evidência atribuídos pelo GO-SIEVe, detalhes da avaliação de cada especialista pode ser encontrado no APÊNDICE G deste trabalho.

Dos 39 termos computados somente três tiveram índice de acerto médio inferior a 50%, porém, mesmo com menor índice de acerto que os demais termos, estes termos foram encontrados em literatura com informação significante, mesmo que em menor quantidade, sendo eles:

- "*direct assay*" com índice de 50% de acerto;
- "*expression profiling*" com índice de 27,3% de acerto e;
- "*total protein*" com índice de 33,3% de acerto.

		Validação		
Código de Evidencia	Termos Cadastrados no GO-SIEVe	Literatura recuperada	Total validado	% de Acerto
CV=	violaceum	8	8	68,8
CV= Total		**8**	**8**	**68,8**
IDA	binding assay	31	29	87,9
	binding experiment	7	6	100,0
	co-immunoprecipitation	4	4	87,5
	direct assay	2	2	50,0
	enzymatic activity	126	48	82,3
	enzyme assay	34	32	100,0
	expressed in vivo	11	11	86,4
	immunofluorescence	66	64	86,7
	immunolocalization	9	8	81,3
	kinetic analysis	33	33	90,9
IDA Total		**323**	**237**	**85,3**
IEP	expression pattern	100	31	96,8
	expression profiling	11	11	27,3
	pattern of expression	19	18	77,8
	transcript level	34	34	83,8
IEP Total		**164**	**94**	**71,4**
IGC	genomic context	3	3	66,7
	operon organization	8	8	62,5
	operon structure	12	12	91,7
IGC Total		**23**	**23**	**73,6**
IGI	functional complementation	47	45	93,3
	rescue experiment	6	6	66,7
	synthetic lethal	1	1	100,0
IGI Total		**54**	**52**	**86,7**
IMP	deletion mutant	104	62	95,2
	gain of function	2	2	100,0
	knockout experiment	4	4	100,0
	loss of function	15	13	88,5
	mutant phenotype	33	32	82,8
IMP Total		**158**	**113**	**93,3**
ISS	chromosome sequence	7	7	100,0
	gene product	782	456	78,6
	genome sequence	268	171	85,7
	genomic sequence	162	162	83,0
	laser desorption ionization	8	8	87,5
	maldi-tof	8	7	100,0
	protein species	5	5	100,0
	proteome	37	37	94,6
	sequence similarity	284	283	75,6
	structural similarity	29	29	89,7
	the genome of	188	188	79,8
	total protein	9	9	33,3
ISS Total		**1787**	**1362**	**84,0**
Total geral		**2517**	**1889**	**82,1**

Quadro 13 – Margem de acertos da validação agrupados pelos termos de cada código de evidência.

A avaliação feita pelos especialistas ao analisarem os códigos de evidência atribuídos pelo GO-SIEVe sobre a literatura é subjetiva, sendo realizada de acordo com a interpretação que cada especialista dá à literatura, somando-se ainda os seus conhecimentos particulares. Com base nesta avaliação subjetiva os especialistas

julgaram se realmente a literatura trouxe ou não informações que dêem suporte às evidências atribuídas pelo GO-SIEVe (BRYSON et al., 2006). Houve, portanto, pequena diferença no julgamento feito pelos especialistas ao quantificarem erros e acertos do GO-SIEVe. O Quadro 14 resume a avaliação de cada especialista e, o Quadro 21 do APÊNDICE G mostra detalhes da validação, agrupando os erros e acertos para cada termo cadastrado no GO-SIEVe (Quadro 7).

	Acertos	Erros	Total
Especialista 1	1542	347	1889
Especialista 2	1561	328	1889
Total Geral	3103	675	

Quadro 14 - Resumo da avaliação feita pelos especialistas.

Um exemplo da diferença na interpretação da literatura é mostrado para o termo “total protein” no APÊNDICE F, que apresenta nove literaturas avaliadas. Enquanto um especialista interpretou como corretas duas (22%) destas literaturas, o outro interpretou como sendo corretas quatro (44%). Entretanto esta diferença no julgamento feito por cada especialista é mínima, menos de 1,23% para os códigos de evidência atribuídos pelo GO-SIEVe no genoma de *C. violaceum*.

5 DISCUSSÃO

5.1 O GO-SIEVe e a BIBLIOGRAFIA

Analisando a literatura encontrada sobre os códigos de evidência e a forma como eles são utilizados nos projetos genoma e nos software de anotação gênica, percebe-se que houve uma evolução natural no uso das evidências. Os primeiros projetos genoma criados não faziam uso dos códigos de evidência, entretanto, os próprios pesquisadores, percebendo a necessidade de manter uma informação mais consistente sobre a anotação gênica, criaram em seus projetos formas particulares de qualificar os genes, geralmente usando categorias de palavras, letras ou números. Apesar das categorias criadas terem um significado particular a cada projeto genoma, para os outros pesquisadores usuários da informação, elas eram vagas, pois forneciam uma informação de um gene, do resultado de uma anotação, entretanto, não forneciam o conhecimento de "como" a informação foi obtida, dos processos envolvidos ao categorizar um gene (BAILEY et al., 1998), (TAKAMI et al., 2000), (STOVER et al., 2000), (AL-LAZIKANI; SHEINERMAN; HONIG, 2001), (KASUKAWA et al., 2003), (SEARLE et al., 2004), (ALMEIDA et al., 2004a), (REY et al., 2004), (BRYSON et al., 2006).

Com o grande volume de informação genética disponível, tornar evidente como é executada uma anotação em particular é uma preocupação atual e, os bancos de dados públicos bem como os pesquisadores estão incluindo em seus projetos o uso dos códigos de evidência definidos pelo GO para fornecer esta evidência junto às informações genéticas (BERARDINI et al., 2004), (LIU; HU; WU, 2005), (SAXONOV; BERG; BRUTLAG, 2006), (AUBRY et al., 2006), (XIANG; ZHENG; HE, 2006), (Flybase - Reference Manual G**,** 2008).

Seguindo esta evolução cronológica, o GO-SIEVe, mesmo não sendo uma plataforma de anotação, procura na literatura de um gene por termos que sirvam de evidência para o processo usado na anotação deste gene e, ao encontrar este termo, atribui um código de evidência ao gene, não sendo obrigatório que seja um

código de evidência do GO. Assim, a metodologia do GO-SIEVe pode ser facilmente inserida em processos ou plataformas de anotação ou re-anotação gênica, independente destas plataformas usarem ou não alguma categoria particular para qualificar seus genes. O GO-SIEVe pode ser usado como uma ferramenta auxiliar para os processos de anotação ou re-anotação existentes e, como os códigos de evidência e seus termos podem ser modificados por especialistas, o GO-SIEVe pode ser facilmente adaptado conforme a natureza do projeto, direcionando os especialistas que necessitam revisar uma anotação automática à leitura de textos pré-selecionados, que contenham informações relevantes (BAILEY et al., 1998), (KASUKAWA et al., 2003), (BOURNE; McENTYRE, 2006), (BRYSON et al., 2006).

Diferente de outras ferramentas, o GO-SIEVe, além de inserir junto às informações do gene um código de evidência, insere o(s) termo(s) relacionado(s) a este código de evidência que foi encontrado na literatura, bem como pode anexar ao gene anotado a própria literatura onde tal evidência foi encontrada. Assim, em um processo de anotação ou re-anotação automática que ainda será validado, além de mostrar informações que dão suporte à anotação, direciona o curador na leitura de informações previamente filtradas (BERARDINI et al., 2004).

5.2 O GO-SIEVe e a QUANTIDADE DE LITERATURA

Analisando a quantidade de literatura recuperada para cada termo, curiosidades surgiram sobre os motivos qual um termo teve mais literatura recuperada do que outros, pois, para alguns termos, poucas unidades de literatura foram recuperadas enquanto que para outros, algumas centenas de literatura (Quadro 13). Será que estas diferenças existem porque os alguns termos são mais comuns que os outros? Por existir maior quantidade de literatura para alguns termos e poucas para outros? É uma característica do genoma de *Chromobacterium violaceum?*

Os mesmos termos processados pelo GO-SIEVe foram submetidos para pesquisa no PubMed NCBI e as quantidades de literatura foram comparadas (NCBI PubMed, 2009). Em média o GO-SIEVe recuperou 0,83% da literatura disponível no PubMed, existindo variações significantes nos extremos desta média para alguns termos: os termos *binding experiment* (4,32%)*, operon organization* (14,29%)*, operon structure* (8,05%)*, functional complementation* (4,90%)*, rescue experiment*

(15,79%), *chromosome sequence* (7,07%) e *genome sequence* (5,19%) tiveram índice acima da média; já os termos *co-imunoprecipitation* (0,12%), *immunofluorescence* (0,10%), *expression profiling* (0,03%), *gain of function* (0,05%), *laser desorption ionization* (0,05%), *maldi-tof* (0,14%) e *total protein* (0,05%) tiveram índice abaixo da média. É interessante observar que, para os termos encontrados com índice acima da média, a quantidade de literatura é baixa tanto no GO-SIEVe quanto no PubMEd.

Mesmo existindo coerencia entre a literatura recuperada pelo GO-SIEVe e a literatura do PubMed, a inexistência de um padrão na quantidade de literatura para alguns termos pode ser devido às característica dos genes de *C. violaceum* e a forma como os genes similares encontrados pelo BLAST foram caracterizados, pois na pesquisa submetida ao PubMed não foram diferenciadas as espécies nem genomas *eukaryotic* dos *prokaryotic*. Para satisfazer cientificamente tal curiosidade um estudo mais aprofundado sobre cada termo e sua relação com os genomas seria necessário.

Em tempo, é relevante lembrar que os *hits* do BLAST utilizados pelo GO-SIEVe para recuperar a litartura do PubMed são de 2003, período em que o genoma de *C. violaceum* foi anotado e validado na plataforma SABIÁ, sendo que hoje existe uma quantidade significativamente maior de genomas anotados e de literatura disponível.

5.3 O GO-SIEVe e o PROCESSO DE ANOTAÇÃO GÊNICA

Na era pós-genômica não só o volume de seqüência pura e estrutura de dados é cada vez maior, mas a sua diversidade também é crescente, levando a um crescimento desproporcional no número de produtos genéticos descaracterizados. Por consequência, métodos criados para anotação de genes e proteínas, como a transferência de homologia, estão anotando menos dados e, em muitos casos, estão ampliando erros existentes em anotações (FRIEDBERG, 2006). No entanto, devido a grande quantidade de seqüência para ser analisada e o volume de dados gerados, o sequenciamento de genomas deve ser automaticamente processado e cuidadosamente filtrado (MUNGALL et al., 2002).

Mais de 400 seqüências de genomas de bactérias estão agora à disposição do público. Um melhor conhecimento da biologia das bactérias depende da

conversão desta matéria-prima em informação, que envolve a identificação e anotação de genes, proteínas e *pathways*. Este processamento é normalmente feito utilizando plataformas de anotação composta de uma variedade de módulos de software e, em alguns casos, especialistas humanos. As bases de dados, conhecimentos e métodos computacionais, que formam a base destes processos estão constantemente evoluindo e, portanto, há necessidade de reprocessar as anotações gênicas regularmente. O desafio de revisar as anotações e extrair informações úteis a partir da quantidade de novas seqüências gênicas exigirá mais confiança em sistemas totalmente automatizados (STOTHARD; WISHART, 2006). A plataforma de anotação SABIÁ (ALMEIDA et al., 2004b) foi a primeira usada para anotar o genoma de *Chromobacterium violaceum* pelo Consórcio Nacional Projeto Genoma Brasileiro (VASCONCELOS et al., 2003). A necessidade em rever a seqüência deste genoma levou ao desenvolvimento de um módulo de software adicional, descritos neste trabalho. Como os códigos de evidência não foram utilizados no texto original da anotação, a adição de uma ferramenta que deduz esses códigos pode facilitar re-anotação. Observando a informação disponível para o anotador na etapa de anotação manual da plataforma SABIÁ e atribuindo os códigos de evidência definidos pelo GO pode ser o ponto de partida para a revisão da anotação original.

A missão do projeto de anotação GO (GOA) é de anotar os genomas e diversas seqüências e estrutura de bases de dados usando os termos GO (CAMON et al., 2004). Quando os termos GO são atribuídos a um produto genético, um código de evidência indica como a anotação foi obtida e como foi inferida também. Desta forma, a confiabilidade da anotação é observada – É a anotação baseada em evidências experimentais que possam ser rastreadas para um autor (alta confiabilidade)? Ou, a anotação foi simplesmente inferida por transferência de homologia baseada em anotações que não tenha sido analisada por um curador (baixa confiabilidade)? O GO-SIEVe utiliza termos pré-definidos contidos na literatura associada a cada gene anotado para atribuir automaticamente os códigos de evidência.

Trabalhando com a anotação de cDNA, Kasukawa et al. (2003) após analisar o padrão humanos de verificação, sugeriu que uma anotação preliminar bem realizada pode, em muitos casos, reduzir a necessidade de intervenção humana. Se anotações não informadas, proteínas hipotéticas, ESTs desconhecidos

e seqüências não classificadas forem excluídas, o processo de anotação automática é aceito pelos curadores em mais de 85% dos casos (KASUKAWA et al., 2003). Similarmente, quando os códigos de evidências foram atribuídos automaticamente pelo GO-SIEVe, os curadores aceitaram 82% das inferências, sugerindo que novas ferramentas automáticas podem ser criadas para melhorar a precisão da anotação.

Não escapou à nossa atenção que, usando os resultados BLAST como fonte da pesquisa bibliográfica para atribuir os códigos de evidência, esta versão do GO-SIEVe esta somente baseada na similaridade da seqüência. Claramente, a similaridade de seqüência esta correlacionada à similaridade funcional, mas exceções são observadas em ambos os extremos desta escala similar. Erros nas anotações podem ser causados porque correspondências na base de dados (*hits)* com significante *e-value* pode ocorrer, mas o gene consultado e as correspondências podem ter um diferente domínio estrutural (FRIEDBERG, 2006). Além disso, Valência (2005) alerta para o fato de que uma anotação funcional atual é baseada essencialmente na expansão de um número relativamente pequeno de funções determinadas experimentalmente para grandes coleções de proteínas (VALENCIA, 2005). A tarefa de anotação sistemática enfrenta problemas práticos relacionados com a exatidão da entrada de informação experimental, com a confiabilidade dos atuais sistemas de transferência de informações entre seqüências relacionadas, e com a reprodutibilidade das ligações entre as informações nas bases de dados e os experimentos originais reportados nas publicações. A simplicidade dos processos de anotações atuais é uma fonte adicional de inquietação quanto à exatidão das funções extrapoladas. Atribuindo automaticamente os códigos de evidência e ligando-os rapidamente à literatura relacionada a eles, pode, além de acelerar o processo de anotação, contribuir para uma maior precisão se mais tempo for deixado para os curadores examinarem cuidadosamente a literatura relacionada e tomar melhores decisões quanto a função de um gene.

5.4 O GO-SIEVe e a ABORDAGEM DE DESENVOLVIMENTO

Quanto ao seu desenvolvimento, O GO-SIEVe poderia ter sido desenvolvido em um único programa ou em uma única classe. Ao invés disto, foi desenvolvido em cinco classes, tornando cada classe, especializada em executar uma tarefa específica, possibilitando que elas sejam reutilizadas de forma independente. As

classes Mensagem e Excecao são classes de apoio ao GO-SIEVe e, poderiam ser reutilizadas em outros programas. A classe ProcessaArquivo poderia ser reutilizada em outro software para fazer *download* de qualquer arquivo da *internet,* sem a necessidade de que as outras classes do GO-SIEVe estivessem integradas neste software. Assim como as classes SabiatoBlast, ProcessaBlast e ProcessaNcbi poderiam ser reutilizadas de forma independente em outros softwares para acessar os genes válidos da plataforma SABIÁ, analisar um arquivo do BLAST ou acessar a literatura encontrada no NCBI respectivamente.

O GO-SIEVe foi desenvolvido com maior preocupação nos processos científicos, na exatidão dos resultados e em sua validação, sendo desconsiderado o desenvolvimento de uma interface amigável com o usuário para a execução do software. Em uma situação de utilização real, onde a metodologia do GO-SIEVe estivesse adaptada em uma plataforma de anotação em uso, as interfaces deveriam ser desenvolvidas em conformidade com o modelo desta plataforma, permitindo uma interação fácil do usuário com a plataforma e o GO-SIEVe. Uma interface gráfica permitiria facilmente o cadastramento dos códigos de evidência e termos de interesse para o projeto e, até mesmo verificar os resultados atribuídos a cada gene. Mesmo sendo possível fazer tudo isto na forma como está o GO-SIEVe atualmente, a execução de tais tarefas dependem de conhecimentos da área de informática.

Para efeito de validação dos resultados e facilidade na verificação dos mesmos, toda a literatura referente aos genes de *C. violaceum* foi armazenada na base de dados do GO-SIEVe, permitindo um acesso mais rápido e independente da internet a estas informações. Entretanto, outra abordagem poderia ser usada, de forma a acessar a literatura na internet e atribuir os códigos de evidência sem que toda literatura fosse armazenada na base de dados local. Algumas considerações entre estas duas abordagens devem ser observadas:

- a literatura pode mudar com a adição de novos artigos no NCBI e, independente de armazenar ou não, um identificador deveria ser criado de forma a reconhecer a literatura já processada ou não pelo GO-SIEVe, sendo que isto poderia ser feito com a utilização do título da literatura ou mesmo o identificador do PubMed para a literatura;
- sendo a literatura armazenada em base de dados local, ao haver mudança na literatura, o GO-SIEVe deveria ser

executado a fim de recuperar, armazenar e atribuir os códigos de evidência nesta nova literatura adicionada;

- sendo a literatura processada on-line, sem armazenamento, não haveria redundância da literatura, porém, atualizações na literatura deveriam ser identificadas de forma a permitir que GO-SIEVe atribuísse os códigos de evidência somente na literatura adicionada, e não em sua totalidade.

Outro aspecto deve ser considerado ao decidir se a literatura será ou não armazenada em base de dados local: a inserção de novos termos ou novos códigos de evidências ou, alteração de algum termo ou código de evidência na base de dados do GO-SIEVe. Neste caso as conseqüências de se armazenar ou não a literatura têm maior importância, pois, havendo alteração de algum termo ou código de evidência, toda literatura deveria ser re-processada em busca destes termos inseridos e alterados, surgindo então duas situações:

- sendo a literatura processada on-line, sem armazenamento, toda a literatura disponível pelo BLAST deveria ser acessada on-line em busca destes novos termos e códigos de evidência, o que consumiria um tempo considerável;
- sendo a literatura armazenada na base de dados do GO-SIEVe, bastaria re-processar e buscar nela os novos termos e códigos de evidência, um processo relativamente rápido e sem custo de processamento.

Ambas as abordagens, de armazenar ou não a literatura, oferecem vantagens e desvantagens, cabendo a decisão de qual abordagem seguir aos responsáveis pelo projeto ao qual o GO-SIEVe estiver inserido.

De forma geral, a gravação da literatura em base de dados local, além de consumir mais espaço de armazenamento, cujo custo é baixo hoje, cria redundância desta literatura. Em contrapartida, a gravação da literatura em base de dados local facilita o acesso e re-processamento caso seja necessário, sendo esta abordagem indicada quando existir incertezas quanto aos códigos de evidência ou termos adotados, pois, se estes forem alterados, inseridos ou excluídos, seria fácil reprocessar toda a literatura e fazer nova atribuição dos códigos de evidência. Quando o projeto tiver certa maturidade e quando existir a certeza de que não

haverá inserções ou alterações nos termos e códigos de evidência, o processamento on-line poderá ser a melhor opção.

O processo para atribuir os códigos de evidência foi programado diretamente na base de dados do GO-SIEVe, onde foi usado apenas recursos do SGBD. Existe, porém, outras técnicas da área da Ciência da Computação que poderiam ser usadas para localizar os termos na literatura e atribuir os códigos de evidência correspondentes, como técnicas do campo de Inteligência Artificial (IA) e Processamento de Linguagem Natural (PLN), cuja eficiência de seus algoritmos em relação ao usado no GO-SIEVe, poderia ser medida em um trabalho científico.

Neste trabalho, foram usados no GO-SIEVe sete códigos de evidência com seus termos relacionados. Tecnicamente o GO-SIEVe permite o uso de um número ilimitado de códigos de evidência e termos. Como os códigos de evidência e seus termos foram construídos na base de dados do GO-SIEVe, e não diretamente no programa, a qualquer momento novos termos podem ser inseridos, alterados ou eliminados na base de dados. Em uma situação real, tanto os códigos de evidência quanto seus termos devem ser registrados conforme a natureza do projeto onde o GO-SIEVe estiver inserido sendo que, para facilitar esta tarefa, uma interface gráfica poderia ser construída para melhorar a interação com os usuários.

6 CONCLUSÃO

O GO-SIEVe, até mesmo por depender de informações de outros softwares (SABIÁ e BLAST), não é uma ferramenta de anotação ou re-anotação completa. Todavia, pode ser usado como um importante módulo ou ferramenta complementar às plataformas de anotação existentes que, usando as informações destas plataformas, pode filtrar a literatura existente e minimizar a quantidade de dados que um curador teria que verificar para validar e atribuir uma função a um gene e, por conseqüência, reduzir tempo e custo envolvidos na anotação ou re-anotação de um genoma.

O GO-SIEVe é uma ferramenta útil para ser usada após a anotação automática de um genoma e antes que os curadores validem esta anotação, principalmente por ser a etapa de leitura da literatura, executada de forma não automatizada, podendo ser otimizada pelo direcionamento da leitura a artigos relevantes.

Observa-se que a maioria da literatura recuperada e conseqüentemente dos códigos de evidências atribuídos a elas foram para os genes categorizados com válidos no genoma de *C. violaceum* (Quadro 8), correspondendo ao processo de validação, onde os curadores encontraram mais informações para validarem os genes.

O GO-SIEVe, ao não encontrar qualquer evidência em algumas literaturas, eliminou 41.9% (75.894) da literatura recuperada que seria lida pelos curadores na anotação do genoma de *C. violaceum*. Para os 58,1% (105.204) restante da literatura que algum código de evidência foi atribuído, o GO-SIEVe acertou em média 82,1%, mostrando-se eficaz na atribuição dos códigos de evidência.

De forma geral, o índice de acerto foi elevado para todos os termos, entretanto, alguns termos tiveram índice de acerto acima da média, como *“binding experiment”, “enzime assay”, “synthetical lethal”, “gain of function”, “knockout experiment”, “chromosome sequence”, “maldi-tof”,* e *“protein species”* entre outros (Quadro 13). Para estes termos foi observado que a literatura recuperada do BLAST

não é abundante, existindo em pouca quantidade, indicando, porém, relevância conforme a evidência atribuída.

Analisando os resultados, é possível observar que o volume de literatura recuperada pelo GO-SIEVe repete-se bastante para todos os códigos de evidência, especialmente para o “ISS”, para o qual somente 1,5% de toda literatura recuperada é única. Do total de 116.182 literaturas, contando as que se repetem *n* vezes como uma unidade, somente 1.787 literaturas são únicas. O segundo código de evidência que possui mais literatura com repetição é o “IDA”, onde somente 8,5% da literatura é única (Quadro 11).

Considerando a contagem da literatura repetida pelos termos cadastrados ao invés dos códigos de evidência, é notável que a literatura para alguns termos repetem-se exageradamente acima da média, são eles: *total protein* (0,8% de literatura única), *the genome off* (0,9%), *proteome* (1,3%), *genomic sequence* (1,2%), *genome sequence* (0,4%) e *chromosome sequence* (0,3%), todos do código de evidência “ISS” e; *expressed in vivo* (0,5%) do código de evidência IDA (Quadro 19 - APÊNDICE D). Somente para o termo *genome sequence*, que é comum na área de genética, foram recuperadas 68.832 literaturas, correspondendo a 56,8% de toda literatura recuperada pelo GO-SIEVe, sendo que, deste total, somente 268 literaturas são únicas, 10,6% do total de literatura única.

Com este grande número de repetição, é natural questionar qual é realmente a relevância do termo e do código de evidência no sentido de revelar uma literatura com informação útil. Sobre este aspecto, duas conclusões podem ser tiradas:

- enquanto método, o GO-SIEVe acertou 82,1% de todas as atribuições, isto quer dizer que, com os termos cadastrados, em média, 82,1% de toda literatura recuperada, seguramente revela informação correspondentes ao código de evidência;
- enquanto relevância do código de evidência no sentido de indicar se uma informação é útil no contexto de anotação gênica, depende do próprio código de evidência (ou do termo relacionado) pois, uma evidência como ISS é menos relevante do que IEP ou IDA, por exemplo e, o GO-SIEVe não tem a capacidade de medir a relevância entre um código de evidência e outro.

A relevância da literatura encontrada é indicada pelo GO-SIEVe pelos códigos de evidência atribuídos, os quais dependem dos termos cadastrados. Então, conlui-se que, quanto mais relevantes forem os termos cadastrados, mais relevantes serão as literaturas sinalizadas com a atribuição de algum código de evidência pelo GO-SIEVe, com índice de 82,1% de acerto.

REFERÊNCIAS

Al-LAZIKANI, Bussan; SHEINERMAN, Felix B.; HONIG, Barry. Combining multiple structure and sequence alignments to improve sequence detection and alignment: Application to the SH2 domain of Janus Kinases. **Proc Natl Acad Sci U S A.** v. 98, n.26, p. 14796-801, dez. 2001.

ALMEIDA, Luiz G. P. et. al. A new set of bioinformatics tools for genome projects. **Genet Mol Res**. v. 3, n. 1, p. 26-52, mar. 2004.

ALMEIDA, Luiz G. P. et. al. A system for Automated Bacterial (genome) Integrated Annotation – SABIÁ. **Bioinformatics**. v. 20, n. 16, p. 2832-33, abr. 2004.

ASHBURNER, Michael et. al. Gene Ontology: tool for the unification of biology, The Gene ontology Consortium. **Nature**. v. 25, p. 25-29, mai. 2000.

AUBRY, Marc et. al. Combining evidence, biomedical literature and statistical dependence: new insights for functional annotation of gene sets. **BMC Bioinformatics**. v. 7, n. 1, p. 241-58, mai. 2006.

BAILEY JR, L. Charles et. al. GAIA: Framework Annotation of Genomic Sequence. **Genome Res**. v. 8, p. 234-50, 1998.

BERARDINI, Tanya Z. et. al. Functional Annotation of the Arabidopsis Genome Using Controlled Vocabularies. **Plant Physiology**. v. 135, p. 745-55, jun. 2004.

BERGERON, Bryan. **Bioinformatics Computing.** Prentice Hall PTR: nov. 2002, 439 p.

BOURNE, Philip E.; McENTYRE, Johanna. Biocurators: Contributors to the World of Science. **Plos**. v. 2, n. 10, p. 1185, out. 2006.

BOOCH, Grady; RUMBAUGH, James; JACOBSON, Ivar. **UML Guia do usuário.** 5. ed. Rio de Janeiro: Capus, 2000. 472 p.

BRUDNO, Michael et al. . Fast and sensitive multiple alignment of large genomic sequence. **BMC Bioinformatics**. v. 4, n.1, p. 66-76, dez. 2003.

BRYSON, K. et. al. AGMIAL: implementing an annotation strategy for prokaryote genomes as a distributed system. **Nucl Acids Res**. v. 34, n.12, p.3533-45, jul. 2006.

CAMARGO, Anamaria A.;, SIMPSON, Andrew J. G.. Collaborative research networks work. **J Clin Invest**. v. 112, n. 4, p. 468-471, aug. 2003.

CAMON, E. et al. The Gene Ontology Annotation (GOA) Database--an integrated resource of GO annotations to the UniProt Knowledgebase. **In Silico Biol**. v. 4, n. 1, p. 5-6, 2004.

CARRARO, Dirce Maria; KITAJIM, João Paulo. Seqüenciamento e Bioinformática de Genomas Bacterianos. **Biotecnolog Ciênc Desenvolv**. n. 28, p. 16-20, out. 2002.

CLAVERIE, Jean-Michel. Do we need a huge new centre to annotate the human genome? **Nature**. v. 403, p. 12, jan. 2000.

ELSIK Christine G. et. al. Community annotation: Procedures, protocols, and supporting tools. **Genome Res**. v. 16, n. 11, p. 1329-33, out. 2006.

European Bioninformatics Institute. **EMBL – Nucleotide Sequence Database.** Disponível em: http://www.ebi.ac.uk/embl/WebFeat/index.html. Acessado em: 23 mai. 2008.

FlyBase – **A Database of *Drosophila* genes and genoma**. Disponível em: <http://flybase.bio.indiana.edu/>. Acessado em: 20 mar. 2008.

Flybase. **Reference Manual G.** Disponível em: <http://flybase.bio.indiana.edu/static_pages/docs/refman/refman-G.html#G.3.2.>. Atualizado em: 21 mai. 2007. Acessado em: 21 mar. 2008.

Friedberg, I. Automated protein function prediction--the genomic challenge. **Brief Bioinform**. v. 7, n. 3, p. 225-42, sep. 2006.

GRUBER, Thomas R. Toward Principles for the design of ontologies used for knowledge sharing. **International Journal Human-computer Studies**. v. 43, n. 5/6. 1993.

Hendler, J. Communication. Science and the semantic web. **Science**. v. 299, n. 5606, p. 520-1, jan. 2003.

International Human Genome Sequencing Consortium. Finishing the euchromatic sequence of the human genome. **Nature**. v. 431, p. 931-45, out. 2004.

KASUKAWA, Takeya et. al. Development and Evaluation of an Automated Annotation Pipeline and cDNA Annotation System. **Genome Res**. v. 13, p. 1542-1551, 2003.

KERSEY, Paul et. al. Integr8 and Genome Reviews: integrated views of complete genomes and proteomes. **Nucl Acid Res**. v. 33 (suppl_1), p. D297-D302, jan. 2005.

KULIKOVA, Tamara et. al. EMBL Nucleotide Sequence Database in 2006. **Nucl Acids Res**. v. 35, n. 1, p. D16-D20, dez. 2006.

LANDER, E. S et. al. Initial sequencing and analysis of the human genome. **Nature**. v. 409, n. 6822, p. 860-921, feb. 2001.

LEWIS, S. E. et. al. Apollo: a sequence annotation editor. **Genome Biology**. v. 3, n. 12, p. 82.1-82.14, dez 2002.

LLIOPOULOS, Loannis et. al. Evaluation of annotation strategies using an entire genome sequence. **Bioinformatics**. v. 19, n. 6, p. 717-26, nov. 2003.

LIU, Hongfang; HU, Zhang-Zi; WU, Cathy H. DynGO: a tool for visualizing anda mining of Gene Ontology and its associations. **BMC Bioinformatics**. v. 6, 201-10, ago. 2005.

MEYERS, Folker et. al. GenDB–an open source genome annotation system for prokaryote genomes. **Nucl Acids Res**. v. 31, n. 8, p. 2187-95, fev. 2003.

MUNGALL, C. J. et. al. An integrated computational pipeline and database to support whole-genome sequence annotation. **Genome Biology**. v. 3, n. 12, p. 1-11, dez. 2002.

NCBI PubMEd - Disponível em: <http://www.ncbi.nlm.nih.gov/pubmed/>. Acessado em: 11 jan. 2009.

NEERINCX, Pieter B.T.; LAUNISSEN, Jack A. M.. Evolution of web services in bioinformatics. **Briefings in bioinformatics**. v. 6, n. 2, p. 178-188, jun. 2005.

NIELSEN, Pernille; KROGH, Anders. A. Large-scale prokaryotic gene prediction and comparison to genome annotation. **Bioinformatics**. v. 21, n. 24, p. 4322-29, out. 2005.

NORMALIZAÇÃO DE TRABALHOS TÉCNICO-CIENTÍFICOS: trabalhos acadêmicos, monografias de graduação, monografia de pós-graduação, dissertações e teses / Sistema Integrado de Bibliotecas da PUCPR. Biblioteca Central. **NBR 14724** (2005); organização, Richardt, Nadia Ficht; Zenere, Cirineo; Lopes, Adriano. Curitiba, 2007. Disponível em: <http://www.biblioteca.pucpr.br/sibi/normas/index.htm>. Acessado em: 20 mar. 2008.

Object Management Group™ (OMG™). **Unified Modeling Language™**. Disponível em: < http://www.uml.org/>. Atualizado em: 15 jan. 2008. Acessado em: 29 jul. 2008.

OLIVEIRA, Talles Henrique Gonçalves de; SANTOS, Neusa Fernandes dos; BELTRAMINI, Leila Maria. O DNA: uma hipótese histórica. **Revista Brasileira de Ensino de Bioquímica e Biológica Molecular.** São Paulo, n. 1, p. 1-16, dez. 2004.

PÉREZ, Antonio J. et. al. Gene annotation from scientific literature using mappings between keyword systems. **Bioinformatics**. v. 20, n. 13, p. 2084-91, abr. 2004.

PERTSEMLIDIS, Alexander; FONDON III, John W. Having a BLAST with bioinformatics (and avoiding BLAST phemy). **Genome Biology**. v. 2, n. 10, set. 2001. Disponível em: <http://genomebiology.com/2001/2/10/reviews/2002>. Acessado em: 20 mar. 2008.

PostgreSQL. **PostgreSQL Global Development Group**. Disponível em: <http://www.postgresql.org/>. Acessado em: 22 mar. 2008.

PRLIC, Andreas et. al. WILMA – automated annotation of protein sequences. **Bioinformatics**. v. 20, n. 1, p. 127-128, jan. 2004.

REY, Michael W. et. al. Complete genome sequence of the industrial bacterium *Bacillus licheniformis* and comparisons with closely related *Bacilus* species. **Genome Biology**. v. 5, n. 10, p. R77.1-R77.12, set. 2004.

ROGOZIN, Igor B. et al. . Computational approaches for the analysis of gene neighbourhoods in prokaryotic genomes. **Briefings in Bioinformatics**. v. 5, n. 4, p. 131-149, jun. 2004.

SAXONOV, Serge; BERG, Paul; BRUTLAG, Douglas L. A genome-wide analysis of CpG dinucleotides in the human genome distinguishes two distinct classes of promoters. **Proc Natl Acad Sci U S A.** v. 103, n. 5, p. 1412-17, jan. 2006.

SCHWARTZ, R. S. Cracking the Genome: Inside the Race to Unlock Human DNA. **N Briefing in Engl J Med**. Boston. p. 344-862, mar. 2001.

SCOTT, Edgar. Automated Annotation. **Ok INBRE Bioinformatics Bulletin**. University of Oklahoma Health Sciences Center. Oklahoma. set. 2006.

SEARLE, Stephen M. J. et al. . The Other Annotation System. **Genome Res.** v. 14, n. 5, p. 963-70, mai. 2004.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSHAN, S. **Sistema de Banco de Dados**, tradução de Vieira D.. 5. ed. Rio de Janeiro: Elsevier, 2006. 781 p.

SIMPSON, Andrew J. G.; CABALLERO, Otávia L. Projeto Genoma Humano e suas implicações para a saúde humana: visão geral e contribuição brasileira para o projeto. **Bioética**. v. 8, n. 1, p. 89-96, 2000; 8(1):89-96.

SIMPSON, Andrew. J. G. et. al. The genome sequence of the plant pathogen Xylella fastidiosa. **Nature**. v. 406, n. 1, p. 151-157, jul. 2000.

SRDANOVIC, Marko et al.. Critical evaluation of the JDO API for the persistence and portability requirements of complex biological databases. **BMC Bioinformatics**. v. 6, n. 1, p. 5-19, jan. 2005.

STOTHARD, P.; WISHART D. S. Automated bacterial genome analysis and annotation. **Curr Opin Microbiol**. v. 9, n. 5, p. 505-10, oct. 2006.

STOVER. C. K. et. al. Complete genome sequence of *Pseudomonas aeruginosa* PA01, an opportunistic pathogen. **Nature**. v. 406, p. 959-64, ago. 2000.

Sun Microsystems. **Sun Developer Network (SDN) - The source for Java developers.** Disponível em: <http://java.sun.com/>. Acessado em: 22 mar. 2008.

TAKAMI, Hideto et. al. Complete genome sequence of the alkaliphilic bacterium *Bacillus halodurans* and genomic sequence comparison with *Bacillus subtilis*. **Nuc Acids Res**. v. 28, n. 21, p. 4317-31, ago. 2000.

The ENCODE Project Consortium. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. **Nature**. v. 447, p. 799-816, jun. 2007.

The Gene Ontology. **Gene Ontology Home**. Disponível em: <http://www.geneontology.org/index.shtml>. Atualizado em: 19 mar. 2008. Acessado em: 20 mar. 2008.

The Gene Ontology. **Guide to GO Evidence Codes**. Disponível em: <http://www.geneontology.org/GO.evidence.shtml>. Atualizado em: 18 jan. 2008. Acessado em: 21 mar. 2008.

VALENCIA, A. Automatic annotation of protein function. **Curr Opin Struct Biol**. v. 15, n. 3, p. 267-74, jun. 2005.

VASCONCELOS, Ana Tereza Ribeiro de et. al. The complete genome sequence for Chromobacterium violaceum reveals remarkable and exploitable bacterial adaptability. **Proc Natl Acad Sci U S A.** v. 100, n. 20, p. 11660-65, set. 2003

VELOSO, Felipe et. al Large-scale, multi-genome analysis of alternate open reading frames in bacteria and archaea. **Omics**. v. 9, n. 1, p. 91-105, mar. 2005.

WEINEL, Christian; ERMOLAEVA, Maria D.; OUZOUNIS, Christos. PseuRECA: genome annotation and gene context analysis for Pseudomonas aeruginosa PA01. **Bioinformatics**. v. 19, n. 12, p. 1457-60, ago. 2003.

WINSOR, Geoffrey L. et. al. *Pseudomonas aeruginosa* Genome Database and PseudoCAP: faciliting community-based, continually updated, genome annotation. **Nucl Acids Res**. v. 33 (suppl_1), p. D338-43, jan. 2005 Jan.

XIANG, Zuoshuang; ZHENG, Wenjie; HE, Yongqun. BBP: Brucella genome annotation with literature mining and curation. **BMC Bioinformatics**. v. 7, n. 1, p. 347-60, jul. 2006.

DOCUMENTOS CONSULTADOS

CAMON, Evelyn et. al. The Gene Ontology Annotation (GOA) Project: Implementation of GO in SWISS-PROT, TrEMBL, and InterPro. **Genome Res**. v. 13, p. 662-72, mar. 2003.

CERAMI, Ethan G. et al. . cPath: open source software for collecting, storing, and querying biological pathways. **BMC Bioinformatics**.v. 7, n. 1, p. 497-505, nov. 2006.

Congresso Nacional de Genética, 51, Águas de Lindóia. **A que veio e para onde vai a Genômica?** São Paulo, set. 2005.

DEITEL, H. M.; DEITEL, P. J. **Java: como programar**. 6. ed. Tradução Furmankiewicz, Edson. São Paulo: Pearson Prentice Hall, 2005. 1110 p.

GIBAS, Cynthia; JAMBECK Per. **Desenvolvendo Bioinformática.** Revisado por Miranda, Antônio Basílio de; Traduzido por Machado, Cristina de Amorim. Rio de Janeiro: Editora Campus, 2001. 440p.

HENNIG, Steffen; GROTH, Detlef, LEHRACH, Hans. Automated Gene Ontology annotation for anonymous sequence data. **Nucl Acids Res**. v. 31, n. 13, p. 3712-15, abr. 2003.

International Committee of Medical Journal Editors. Uniform Requirements for Manuscripts Submitted to Biomedical Journals: Writing and Editing for Biomedical Publication. Bethesda. 2006; atualizado em fevereiro 2006, disponível em http://www.icmje.org/

KAWAS, Edward; SENGER, Martin; WILKINSON, Mark D.. BioMoby extensions to the Taverna workflow management and enactment software. **BMC Bioinformatics**. v. 7, n. 1, p. 523-535, nov. 2006.

KOIKE, Azako; NIWA, Yoshiki; TAKAGI, Toshihisa. Automatic extraction of gene/protein biological function from biomedical text. **Bioinformatics**. v. 21, n. 7, p. 1227-36, out. 2004.

LEMOS, Melissa; SEIBEL Luiz Fernando Bessa; CASANOVA, Marcos Antônio. Sistemas de Anotação em Bioseqüência. PUC-RIO Inf. MCC04/03. 2003 fev.

Disponível em <ftp://ftp.inf.puc-rio.br/pub/docs/techreports/03_04_lemos.pdf>. Acessado em 15 jun. 2006.

LOMAX, Jane. Get ready to GO! A biologist's guide to the Gene Ontology. **Briefings in Bioinformatics**. v. 6, n. 3, p. 298-304, set. 2005.

PROSDOCIMI, Francisco et al. . Bioinformática: Manual do usuário. **Biotecnolog Ciênc Desenvolv**. Brasília, v. 29, n.1, p. 18-31, jan. 2003.

SHANNON, Paul T. et. al. The Gaggle: An open-source software system for integrating bioinformatics software and data sources. **BMC Bioinformatics**. v. 7, p. 176-88, mar. 2006.

SIMPSON, Andrew J. G. et. al. Coordinated, network-based research as a strategic component of science in Brazil. **Genet Mol Res**. v. 3, n. 1, p. 18-25, mar. 2004.

Swiss-Prot Protein Knowledg. Disponível em <http://wwww.expasy.org/sprot/>. Acesso em 29 dez. 2006.

TAMAMES, Javier. Evolution of gene order conservation in prokaryotes. **Genome Biology**. v. 2, n. 6, p. 20.1-20.11, jun. 2001.

VENTER, J. Craig et al. . The Sequence of the Human Genome. **Science**. v. 291, p. 1304-51, fev. 2001.

WATERSTON, Robert H.; LANDER, Eric S.; SULSTON, John E.. On the sequencing of the human genoma. **Proc Natl Acad Sci U S A**. v. 99, n. 6, p. 3712-16, mar 2003.

WU, Xiaomei et. al. Prediction of yeast protein-protein interaction networks: insights from the Gene ontology and annotations. **Nucl Acids Res**. v. 34, n. 7, p. 2137-50, mar. 2006.

WU, Xiaomei et. al. SPIDer: *Saccharomyces* protein-protein interaction database. **BMC Bioinformatics**. v. 7, n. 5, p. S16-S21, dez. 2006.

GLOSSÁRIO

BLASTn	BLAST que compara e mostra o grau de similaridade entre uma seqüência problema *(query)* e as seqüências no banco de dados *(subject)* com base em nucleotídeos.
BLASTp	BLAST que compara e mostra o grau de similaridade entre uma seqüência problema *(query)* e as seqüências no banco de dados *(subject)* com base em aminoácidos.
Códon	Seqüência de três nucleotídeos que codificam um aminoácido de uma cadeia polipeptídica (proteína).
Genoma	Informação genética total carregada por uma célula ou organismo
Gb	Simbologia usada para representar os genes depositados nas bases de dados públicas, quais serão procurados em uma busca por similaridade (*subject*).
Gi	Simbologia usada para representar um gene de interesse em uma busca por similaridade (*query*).
Motif	Elemento de estrutura ou padrão repetido em diferentes proteínas.

APÊNDICES

APÊNDICE A - Quantidade de literaturas armazenadas por dia

Data	Quantidade
18-out-07	577
19-out-07	2817
20-out-07	4844
21-out-07	3543
22-out-07	4747
23-out-07	4358
24-out-07	4608
25-out-07	4176
26-out-07	3281
27-out-07	3222
28-out-07	4200
29-out-07	3384
30-out-07	3283
31-out-07	4386
1-nov-07	1434
13-nov-07	304
14-nov-07	1626
22-nov-07	2126
23-nov-07	1700
24-nov-07	4493
25-nov-07	2923
30-nov-07	2745
1-dez-07	1463
2-dez-07	3772
3-dez-07	3395
4-dez-07	4730
5-dez-07	1848
14-dez-07	3332
15-dez-07	682
16-dez-07	5795
17-dez-07	6731
18-dez-07	4561
19-dez-07	6003
20-dez-07	1988
21-dez-07	4796
22-dez-07	5172
23-dez-07	3659
18-jan-07	4116
19-jan-07	1867
25-jan-07	3213
31-jan-08	4187
1-fev-08	5061
2-fev-08	4784
3-fev-08	6301
4-fev-08	4505
5-fev-08	6219
6-fev-08	4981
7-fev-08	2580
8-fev-08	6580
TOTAL	**181098**
MÉDIA	**3695,877551**

Quadro 15 - Quantidade de literatura armazenada diariamente.

APÊNDICE B – Quantidade de literatura por códigos de evidência

Evidências	Literatura	:CV
:CV=:IMP	1	
:IGC:IGC	1	
:IGC:IMP:ISS	1	
:IGI:IMP:ISS	1	1
:IGI:IGI	1	1
:IDA:IGI:ISS	1	1
:IDA:IGI	2	2
:IEP:IMP:ISS	2	
:IEP:IGI	2	2
:IGI:IMP	3	3
:IDA:IMP	3	
:IDA:IGC:IGI:ISS	4	4
:IDA:IDA	4	
:CV=	5	
:IDA:ISS:ISS	5	
:IEP:IMP	6	
:IDA:IEP	7	
:CV=:CV=	9	
:IGC:ISS	11	
:IEP:ISS	14	
:IEP:IEP	18	
:IMP:IMP	26	
:IGI:ISS	32	32
:IMP:ISS	67	
:IGI	80	80
:IGC	165	
:IEP	316	
:IMP	326	
:IDA:ISS	1086	
:IDA	2692	
:ISS:ISS	44276	
:ISS	56037	
Inferências	**105204**	**126**
Sem evidência	75894	
	181098	

Quadro 16 - Quantidade de literatura sumarizada pelos agrupamentos dos códigos de evidência.

APÊNDICE C – Códigos de evidência e os termos com a quantidade de atribuição

Código Evidência	Termos	Quantidade de atribuição
IMP	knock-out experiment	2
IMP	gain of function	3
IMP	knockout experiment	4
IDA	direct assay	5
IDA	co-immunoprecipitation	6
IGC	genomic context	7
ISS	protein species	8
IGI	synthetic lethal	8
IGI	rescue experiment	8
IDA	binding experiment	8
ISS	laser desorption ionization	10
IDA	immunolocalization	11
ISS	maldi-tof	15
CV=	violaceum	15
IGC	operon structure	26
IMP	loss of function	28
IEP	pattern of expression	34
ISS	structural similarity	50
IEP	transcript level	51
IDA	enzyme assay	76
IDA	binding assay	90
IMP	mutant phenotype	90
IDA	immunofluorescence	101
IGI	functional complementation	110
IEP	expression profiling	127
IGC	operon organization	149
IEP	expression pattern	158
IDA	enzymatic activity	236
IMP	deletion mutant	309
IDA	kinetic analysis	1045
ISS	total protein	1083
ISS	chromosome sequence	2212
IDA	expressed in vivo	2226
ISS	proteome	2786
ISS	sequence similarity	2826
ISS	gene product	4597
ISS	genomic sequence	13105
ISS	the genome of	20658
ISS	genome sequence	68832
TOTAL		**121115**

Quadro 17 - Quantidade de atribuições por termo cadastrado, relacionados a seus respectivos códigos de evidência.

APÊNDICE D – Repetição na Literatura recuperada pelo BLAST

Evidencia	Termo	Repetição	Título do Artigo
ISS	genome sequence	4279	Comparison of the genomes of two Xanthomonas pathogens with differing host specificities
ISS	genome sequence	3779	Complete genome sequence of Pseudom+D3onas aeruginosa PA01, an opportunistic pathogen
ISS	genome sequence	3153	Complete genome sequence and comparative analysis of the metabolically versatile Pseudomonas putida
ISS	genome sequence	3089	Genome sequence of the plant pathogen Ralstonia solanacearum
ISS	genomic sequence	2307	Complete genomic sequence of nitrogen-fixing symbiotic bacterium Bradyrhizobium japonicum USDA110
ISS	the genome of	2307	Complete genomic sequence of nitrogen-fixing symbiotic bacterium Bradyrhizobium japonicum USDA110
ISS	genomic sequence	2306	Complete genomic sequence of nitrogen-fixing symbiotic bacterium Bradyrhizobium japonicum USDA110 (
IDA	expressed in vivo	2201	Characterization and Pathogenic Significance of Vibrio vulnificus Antigens Preferentially Expressed in Sept
ISS	genome sequence	2075	Genome sequence of the dissimilatory metal ion-reducing bacterium Shewanella oneidensis
ISS	the genome of	2051	Complete genome structure of the nitrogen-fixing symbiotic bacterium Mesorhizobium loti
ISS	genome sequence	2041	Genome sequence of enterohaemorrhagic Escherichia coli O157:H7
ISS	the genome of	2041	Genome sequence of enterohaemorrhagic Escherichia coli O157:H7
ISS	genome sequence	2038	Reannotation of Shewanella oneidensis genome
ISS	genome sequence	1994	Complete genome sequence of a multiple drug resistant Salmonella enterica serovar Typhi CT18
ISS	genome sequence	1956	The Brucella suis genome reveals fundamental similarities between animal and plant pathogens and symb
ISS	genomic sequence	1955	DNA sequence of both chromosomes of the cholera pathogen Vibrio cholerae
ISS	genome sequence	1945	The complete genome sequence of Escherichia coli K-12
ISS	genome sequence	1891	Genome sequence of Yersinia pestis, the causative agent of plague
ISS	genomic sequence	1858	Annotation and evolutionary relationships of a small regulatory RNA gene micF and its target ompF in Yers
ISS	genome sequence	1650	Genome sequence of the plant pathogen and biotechnology agent Agrobacterium tumefaciens C58
ISS	the genome of	1650	Genome sequence of the plant pathogen and biotechnology agent Agrobacterium tumefaciens C58
ISS	genome sequence	1512	Extensive mosaic structure revealed by the complete genome sequence of uropathogenic Escherichia coli
ISS	genome sequence	1508	Complete genome sequence of Salmonella enterica serovar Typhimurium LT2
ISS	genome sequence	1480	The composite genome of the legume symbiont Sinorhizobium meliloti
ISS	genome sequence	1466	Escherichia coli K-12: a cooperatively developed annotation snapshot--2005
ISS	genome sequence	1458	Complete genome sequence of the model actinomycete Streptomyces coelicolor A3(2)
ISS	genome sequence	1331	and O157
ISS	the genome of	1331	and O157
ISS	genome sequence	1301	Complete genome sequence of Caulobacter crescentus
ISS	genome sequence	1274	Complete genome sequence of Neisseria meningitidis serogroup B strain MC58
ISS	the genome of	1274	The genome of the natural genetic engineer Agrobacterium tumefaciens C58
ISS	chromosome sequence	1187	Analysis of the chromosome sequence of the legume symbiont Sinorhizobium meliloti strain 1021
ISS	genome sequence	1086	Complete DNA sequence of a serogroup A strain of Neisseria meningitidis Z2491
ISS	genome sequence	1085	The genome sequence of the facultative intracellular pathogen Brucella melitensis
ISS	the genome of	1085	The genome sequence of the facultative intracellular pathogen Brucella melitensis
ISS	genome sequence	1078	Sequencing and Analysis
ISS	proteome	1072	Proteome analysis of Neisseria meningitidis serogroup A
ISS	total protein	1072	Proteome analysis of Neisseria meningitidis serogroup A
ISS	genome sequence	1059	Comparative genomics of Listeria species
IDA	kinetic analysis	998	Discovery, characterization and kinetic analysis of an alditol oxidase from streptomyces coelicolor
ISS	the genome of	998	Discovery, characterization and kinetic analysis of an alditol oxidase from streptomyces coelicolor
ISS	genome sequence	972	The complete genome sequence of the gram-positive bacterium Bacillus subtilis
ISS	genome sequence	958	Complete genomic sequence of Pasteurella multocida,Pm70
ISS	genomic sequence	958	Complete genomic sequence of Pasteurella multocida,Pm70
ISS	the genome of	944	Sequencing of three lambda clones from the genome of alkaliphilic Bacillus sp. strain C-125
ISS	genome sequence	937	subtilis
ISS	the genome of	937	An improved physical and genetic map of the genome of alkaliphilic Bacillus sp. C-125
ISS	gene product	935	Characterization and comparative study of the rrn operons of alkaliphilic Bacillus halodurans C-125
ISS	sequence similarity	935	Sequence analysis of a 32-kb region including the major ribosomal protein gene clusters from alkaliphilic B
ISS	gene product	934	Replication origin region of the chromosome of alkaliphilic Bacillus halodurans C-125
ISS	the genome of	933	Analysis of the genome of an alkaliphilic Bacillus strain from an industrial point of view
ISS	genome sequence	912	Whole-genome random sequencing and assembly of Haemophilus influenzae Rd
ISS	genome sequence	845	fastidiosa
ISS	genomic sequence	839	Complete genomic sequence of the filamentous nitrogen-fixing cyanobacterium Anabaena sp. strain PCC
ISS	the genome of	839	Complete genomic sequence of the filamentous nitrogen-fixing cyanobacterium Anabaena sp. strain PCC
ISS	gene product	818	Two-dimensional map of the proteome of Haemophilus influenzae
ISS	proteome	818	Two-dimensional map of the proteome of Haemophilus influenzae
ISS	genome sequence	811	Metabolism and evolution of Haemophilus influenzae deduced from a whole-genome comparison with Esch
ISS	genome sequence	744	Frequency and distribution of DNA uptake signal sequences in the Haemophilus influenzae Rd genome
ISS	proteome	744	Reference map of the low molecular mass proteins of Haemophilus influenzae
		90034	

Quadro 18 - Título e quantidade das 60 literaturas que mais se repetem.

Código Evidencia	Termo	Literatura Recuperada	Literatura única	% Literatura única
CV=	violaceum	15	8	53,3
CV= Total		**15**	**8**	**53,3**
IDA	binding assay	90	31	34,4
	binding experiment	8	7	87,5
	co-immunoprecipitation	6	4	66,7
	direct assay	5	2	40,0
	enzymatic activity	236	126	53,4
	enzyme assay	76	34	44,7
	expressed in vivo	2226	11	0,5
	immunofluorescence	101	66	65,3
	immunolocalization	11	9	81,8
	kinetic analysis	1045	33	3,2
IDA Total		**3804**	**323**	**8,5**
IEP	expression pattern	158	100	63,3
	expression profiling	127	11	8,7
	pattern of expression	34	19	55,9
	transcript level	51	34	66,7
IEP Total		**370**	**164**	**44,3**
IGC	genomic context	7	3	42,9
	operon organization	149	8	5,4
	operon structure	26	12	46,2
IGC Total		**182**	**23**	**12,6**
IGI	functional complementation	110	47	42,7
	rescue experiment	8	6	75,0
	synthetic lethal	8	1	12,5
IGI Total		**126**	**54**	**42,9**
IMP	deletion mutant	309	104	33,7
	gain of function	3	2	66,7
	knockout experiment	4	2	50,0
	knock-out experiment	2	2	100,0
	loss of function	28	15	53,6
	mutant phenotype	90	33	36,7
IMP Total		**436**	**158**	**36,2**
ISS	chromosome sequence	2212	7	0,3
	gene product	4597	782	17,0
	genome sequence	68832	268	0,4
	genomic sequence	13105	162	1,2
	laser desorption ionization	10	8	80,0
	maldi-tof	15	8	53,3
	protein species	8	5	62,5
	proteome	2786	37	1,3
	sequence similarity	2826	284	10,0
	structural similarity	50	29	58,0
	the genome of	20658	188	0,9
	total protein	1083	9	0,8
ISS Total		**116182**	**1787**	**1,5**
Total geral		**121115**	**2517**	**2,1**

Quadro 19 - Quantidade de literatura repetida agrupada por termos.

APÊNDICE E – Processamento do gene CV2101 de *C. violaceum*

Código	Categoria	Gene	Evidências	Termos
1925	Y	CV2101		-
1925	Y	CV2101	:ISS:ISS	:the genome of - :genomic sequence
1925	Y	CV2101		-
1925	Y	CV2101	:ISS	- :genome sequence
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101	:ISS	- :genome sequence
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:ISS:ISS	:the genome of - :genome sequence
1925	Y	CV2101		-
1925	Y	CV2101	:ISS	- :the genome of
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101	:ISS	- :genome sequence
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101		-
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101	:IMP	:deletion mutant -
1925	Y	CV2101		-
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:ISS	:genome sequence -
1925	Y	CV2101	:ISS	- :the genome of
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101	:ISS	- :genomic sequence
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101		-
1925	Y	CV2101	:ISS	:the genome of -
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101	:ISS	:sequence similarity -
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101		-
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:IEP	:expression pattern -
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101	:ISS:ISS	:the genome of - :genome sequence
1925	Y	CV2101	:IDA:ISS	:the genome of - :kinetic analysis
1925	Y	CV2101		-
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101		-
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101		-
1925	Y	CV2101	:ISS	- :chromosome sequence
1925	Y	CV2101	:ISS	:genome sequence -
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence
1925	Y	CV2101	:ISS	:genome sequence -
1925	Y	CV2101	:ISS	:genome sequence -
1925	Y	CV2101		-
1925	Y	CV2101	:ISS:ISS	:genome sequence - :genome sequence

Quadro 20 - Listagem de todos os códigos de evidência atribuídos ao gene CV2101

APÊNDICE F – Literatura Encontrada para o Termo "*Total Protein*"

TÍTULO: Cloning, expression and mutational analysis of the urocanase gene (hutU) from Pseudomonas putida. RESUMO: The histidine-utilizing hutU gene was isolated from a lambda-EMBL3 phage of a genomic library from Pseudomonas putida nicII and subcloned into the expression vector pT7-7. Escherichia coli BL21 cells were transformed with the recombinant plasmid and produced a catalytically active protein, amounting to approximately 30% of the **total protein** in the crude cell-free extract. The addition of NAD+ to the growth medium ensured the full occupation of active sites by the cofactor. This requires a mechanism for the transport of NAD+ into E. coli cells. Using the overproducing mutant a new, fast and efficient isolation procedure is described which yields electrophoretically homogeneous urocanase within two days. The yield of pure enzyme, based on the culture volume, has been improved 50-80-fold compared with the traditional method. To investigate the possible role of cysteine residues in the catalysis or in the tight binding of the cofactor NAD+, six different mutants were prepared. In each mutant protein, one conserved cysteine was exchanged for alanine. The resulting clones were tested for the expression of urocanase with catalytic activity; the Km and Vmax values were determined. Only Cys410 was essential for catalysis. There was no detectable reconstitution or increase of activity after the addition of NAD+, either in the essential Cys/Ala mutant or the other mutant proteins. Electrospray-mass spectroscopy of the wild-type enzyme revealed that the coenzyme is not covalently bound to the protein and computational analysis showed no typical sequence for a mononucleotide-binding domain like the Rossman fold. To obtain urocanase apoenzyme, P. putida nicII was transformed with pGP1-2 and pTET7-U and grown in nicotinate-depleted medium. Like the mutant proteins, no activation of the apoform occurred after the addition of NAD+. These observations led us to postulate a new model for the non-covalent but tight binding of NAD+ to the enzyme by *trapping* the cofactor while folding the nascent protein.
TÍTULO: Dynamics of endogenous ATP7A (Menkes protein) in intestinal epithelial cells: copper-dependent redistribution between two intracellular sites. RESUMO: We report for the first time on the copper-dependent behavior of endogenous ATP7A in two types of polarized intestinal epithelia, rat enterocytes in vivo and filter-grown Caco-2 cells, an accepted in vitro model of human small intestine. We used high-resolution, confocal immunofluorescence combined with quantitative cell surface biotinylation and found that the vast majority of endogenous ATP7A was localized intracellularly under all copper conditions. In copper-depleted cells, virtually all of the ATP7A localized to a post-TGN compartment, with <3% of the **total protein** detectable at the basolateral cell surface. When copper levels were elevated, ATP7A dispersed to the cell periphery in punctae whose pattern did not overlap with the steady-state distributions of post-Golgi, endosomal, or basolateral membrane markers; only approximately 8-10% of the recovered ATP7A was detected at the basolateral cell surface. These results raise several questions regarding prevailing models of ATP7A dynamics and the mechanism of copper efflux.
TÍTULO:Expression of a thioredoxin peroxidase in insulin-producing cells RESUMO:The presence of thioredoxin peroxidase (TPx), also known as thiol specific antioxidant (TSA), was investigated in neonatal and adult rat islets, and in the beta-cell line HIT-T15. Western blotting of extracts from neonatal and adult pancreatic islets and from the tumoral cell line HIT-T15 revealed the presence of a 25 kDa protein that comigrated with purified yeast TPx. Endocrine pancreatic TPx accounted for approximately 0.01% of the **total protein** content. Treatment with H2O2 for 3 h increased the expression of TPx in HIT-T15 cells. The distribution of TPx throughout the islet cells was confirmed by immunocytochemistry. Since pancreatic beta-cells possess a weak antioxidant enzyme defense system, especially with regard to hydrogen peroxidase-decomposing enzymes, the presence of a TPx analog in islets suggests that this enzyme may play a role in protecting pancreatic cells against reactive oxygen species.
TÍTULO: The Unique tuf2 Gene from the Kirromycin Producer Streptomyces ramocissimus Encodes a Minor and Kirromycin-Sensitive Elongation Factor Tu RESUMO: Streptomyces ramocissimus, the producer of elongation factor Tu (EF-Tu)-targeted antibiotic kirromycin, contains three divergent tuf-like genes, with tuf1 encoding regular kirromycin-sensitive EF-Tu1; the functions of tuf2 and tuf3 are unknown. Analysis of the tuf gene organization in nine producers of kirromycin-type antibiotics revealed that they all contain homologues of tuf1 and sometimes of tuf3 but that tuf2 was found in S. ramocissimus only. The tuf2-flanking regions were sequenced, and the two tuf2-surrounding open reading frames were shown to be oriented in opposite directions. In vivo transcription analysis of the tuf2 gene displayed an upstream region with bidirectional promoter activity. The transcription start site of tuf2 was located approximately 290 nucleotides upstream of the coding sequence. Very small amounts of tuf2 transcripts were detected in both liquid- and surface-grown cultures of S. ramocissimus, consistent with the apparent absence of EF-Tu2 in **total protein** extracts. The tuf2 transcript level was not influenced by the addition of

kirromycin to exponentially growing cultures. To assess the function of S. ramocissimus EF-Tu2, the protein was overexpressed in Streptomyces coelicolor LT2. This strain is a J1501 derivative containing His(6)-tagged EF-Tu1 as the sole EF-Tu species, which facilitated the separation of EF-Tu2 from the interfering EF-Tu1. S. ramocissimus EF-Tu1 and EF-Tu2 were indistinguishable in their ability to stimulate protein synthesis in vitro and exhibited the same kirromycin sensitivity, which excludes the possibility that EF-Tu2 is directly involved in the kirromycin resistance mechanism of S. ramocissimus.

TÍTULO: Whole-cell kinetics of trichloroethylene degradation by phenol hydroxylase in a ralstonia eutropha JMP134 derivative.

RESUMO: The rate, progress, and limits of trichloroethylene (TCE) degradation by Ralstonia eutropha AEK301/pYK3021 whole cells were examined in the absence of aromatic induction. At TCE concentrations up to 800 &mgr;M, degradation rates were sustained until TCE was no longer detectable. The Ks and Vmax for TCE degradation by AEK301/pYK3021 whole cells were determined to be 630 &mgr;M and 22.6 nmol/min/mg of **total protein**, respectively. The sustained linear rates of TCE degradation by AEK301/pYK3021 up to a concentration of 800 &mgr;M TCE suggest that solvent effects are limited during the degradation of TCE and that this construct is little affected by the formation of toxic intermediates at the TCE levels and assay duration tested. TCE degradation by this strain is subject to carbon catabolite repression.

TÍTULO: Identification of the 50S ribosomal proteins from the Eubacterium Thermus thermophilus.

RESUMO: The **total protein** mixture from the 50S subunit (TP-50) of the eubacterium Thermus thermophilus was characterized after blotting onto PVDF membranes from two-dimensional polyacrylamide gel electrophoresis (2D-PAGE) and sequencing. The proteins were numbered according to their primary structure similarity with their counterparts from other species. One of them has been marked with an asterisk, namely L*23, because unlike the other known ribosomal proteins it shows a very low degree of homology. A highly acidic 5S rRNA binding protein, TL5, was characterized and compared with the available primary structure information. Proteins L1 and L4 migrate similarly on 2D-PAGE. Protein L4, essential for protein biosynthesis, is N-terminally blocked and shows a strikingly low homology to other L4 proteins. In addition to L4, two other proteins, namely L10 and L11, were found to be N-terminally blocked. In conclusion, 33 proteins from the large subunit were identified, including TL5. Homologs to rpL25 and rpL26 were not found.

TÍTULO: Primary structure of human deoxycytidylate deaminase and overexpression of its functional protein in Escherichia coli.

RESUMO: The cDNA encoding human dCMP deaminase was isolated from a lambda ZAPII expression library using an antibody generated against highly purified HeLa cell dCMP deaminase. The cloned cDNA consists of 1856 base pairs and encodes a protein of 178 amino acids with a calculated molecular mass of 19,985 daltons. The sequence of several cyanogen bromide-cleaved peptides derived from HeLa cell dCMP deaminase are all contained within the deduced amino acid sequence. A zinc binding region is present in the enzyme, similar to that reported for cytidine deaminase (Yang, E. C., Carlow, D., Wolfenden, R., and Short, S. A. (1992) Biochemistry 31, 4168-4174). Northern blot analysis revealed a predominant messenger RNA species of 1.9 kilobases. Expression of the active protein to about 10% of Escherichia coli*s **total protein** was achieved by subcloning the open reading frame into a high expression system using the polymerase chain reaction. Polyacrylamide gel electrophoresis revealed a prominent protein band which comigrated with affinity purified HeLa dCMP deaminase, while Western blot analysis yielded an immunoreactive band which comigrated with the single immunoreactive affinity column purified dCMP deaminase band. The enzyme which possesses a kcat of 1.02 x 10(3) s-1 was purified to homogeneity in over 60% yield. The overexpression of dCMP deaminase should permit more exacting studies on the regulation of this important allosteric enzyme which provides substrate for DNA synthesis.

TÍTULO: Molecular cloning and characterization of apricot fruit polyphenol oxidase

RESUMO: A reverse transcriptase-polymerase chain reaction experiment was done to synthesize a homologous polyphenol oxidase (PPO) probe from apricot (Prunus armeniaca var Bergeron) fruit. This probe was further used to isolate a full-length PPO cDNA, PA-PPO (accession no. AF020786), from an immature-green fruit cDNA library. PA-PPO is 2070 bp long and contains a single open reading frame encoding a PPO precursor peptide of 597 amino acids with a calculated molecular mass of 67.1 kD and an isoelectric point of 6.84. The mature protein has a predicted molecular mass of 56.2 kD and an isoelectric point of 5.84. PA-PPO belongs to a multigene family. The gene is highly expressed in young, immature-green fruit and is turned off early in the ripening process. The ratio of PPO protein to **total protein**s per fruit apparently remains stable regardless of the stage of development, whereas PPO specific activity peaks at the breaker stage. These results suggest that, in addition to a transcriptional control of PPO expression, other regulation factors such as translational and posttranslational controls also occur.

TÍTULO: Proteome analysis of Neisseria meningitidis serogroup A.

RESUMO: Neisseria meningitidis is an encapsulated Gram-negative bacterium responsible for significant morbidity and mortality worldwide. Meningococci are opportunistic pathogens, carried in the nasopharynx of approximately 10% of asymptomatic adults. Occasionally they enter the bloodstream to cause septicaemia and meningitis. Meningococci are classified into serogroups on the basis of polysaccharide capsule diversity, and serogroup A strains have caused major epidemics mainly in the developing world. Here we describe a two-dimensional gel electrophoresis protein map of the serogroup A strain Z4970, a clinical isolate classified as ancestral to several pandemic waves. To our knowledge this is the first systematically annotated proteomic map for N. meningitidis. **Total protein** samples from bacteria grown on GC-agar were electrophoretically separated and protein species were identified by matrix-assisted laser desorption/ionization time of flight spectrometry. We identified the products of 273 genes, covering several functional classes, including 94 proteins so far considered as hypothetical. We also describe several protein species encoded by genes reported by DNA microarray studies as being regulated in physiological conditions which are relevant to natural meningococcal pathogenicity. Since menA differs from other serogroups by having a fairly stable clonal population structure (i.e. with a low degree of variability), we envisaged comparative mapping as a useful tool for microevolution studies, in conjunction with established genotyping methods. As a proof of principle, we performed a comparative analysis on the B subunit of the meningococcal transferrin receptor, a vaccine candidate encoded by the tbpB gene, and a known marker of population diversity in meningococci. The results show that TbpB spot pattern variation observed in the maps of nine clinical isolates from diverse epidemic spreads, fits previous analyses based on allelic variations of the tbpB gene.

APÊNDICE G – Informações Gerais da Validação

TERMO	TOTAL	Validador 01 SIM	NÃO	% Sim	% Não	Validador 02 SIM	NÃO	% Sim	% Não	Média Acerto
Gene Product	456	347	109	76,1	23,9	370	86	81,1	18,9	78,6
Genome Serquence	171	147	24	86,0	14,0	146	25	85,4	14,6	85,7
Genomic Sequence	162	133	29	82,1	17,9	136	26	84,0	16,0	83,0
The genome of	188	153	35	81,4	18,6	147	41	78,2	21,8	79,8
Sequence Similarity	283	212	71	74,9	25,1	216	67	76,3	23,7	75,6
Proteome	37	35	2	94,6	5,4	35	2	94,6	5,4	94,6
Chromosome Sequence	7	7	0	100,0	0,0	7	0	100,0	0,0	100,0
Expression Pattern	31	30	1	96,8	3,2	30	1	96,8	3,2	96,8
Deletion Mutant	62	59	3	95,2	4,8	59	3	95,2	4,8	95,2
Kinetic Analysis	33	30	3	90,9	9,1	30	3	90,9	9,1	90,9
Expressed in vivo	11	9	2	81,8	18,2	10	1	90,9	9,1	86,4
Enzimatic activity	48	40	8	83,3	16,7	39	9	81,3	18,8	82,3
Total protein	9	4	5	44,4	55,6	2	7	22,2	77,8	33,3
Structural similarity	29	27	2	93,1	6,9	25	4	86,2	13,8	89,7
Binding Experiment	6	6	0	100,0	0,0	6	0	100,0	0,0	100,0
Co-immunoprecipitation	4	4	0	100,0	0,0	3	1	75,0	25,0	87,5
Direct Assay	2	1	1	50,0	50,0	1	1	50,0	50,0	50,0
Immunolocalization	8	6	2	75,0	25,0	7	1	87,5	12,5	81,3
Pattern of expression	18	14	4	77,8	22,2	14	4	77,8	22,2	77,8
Transcript level	34	29	5	85,3	14,7	28	6	82,4	17,6	83,8
Genomic context	3	2	1	66,7	33,3	2	1	66,7	33,3	66,7
Operon Structure	12	11	1	91,7	8,3	11	1	91,7	8,3	91,7
Rescue experiment	6	4	2	66,7	33,3	4	2	66,7	33,3	66,7
Synthetic lethal	1	1	0	100,0	0,0	1	0	100,0	0,0	100,0
Gain of function	2	2	0	100,0	0,0	2	0	100,0	0,0	100,0
Knock-out experiment	4	4	0	100,0	0,0	4	0	100,0	0,0	100,0
Loss of function	13	12	1	92,3	7,7	11	2	84,6	15,4	88,5
Laser-desorption ionization	8	7	1	87,5	12,5	7	1	87,5	12,5	87,5
Maldi-Tof	7	7	0	100,0	0,0	7	0	100,0	0,0	100,0
Protein Species	5	5	0	100,0	0,0	5	0	100,0	0,0	100,0
Mutant Phenotype	32	27	5	84,4	15,6	26	6	81,3	18,8	82,8
Binding Assay	29	25	4	86,2	13,8	26	3	89,7	10,3	87,9
Enzyme Assay	32	32	0	100,0	0,0	32	0	100,0	0,0	100,0
Immunofluorescence	64	54	10	84,4	15,6	57	7	89,1	10,9	86,7
Operon organization	8	5	3	62,5	37,5	5	3	62,5	37,5	62,5
Expression Profiling	11	3	8	27,3	72,7	3	8	27,3	72,7	27,3
Functional Complementation	45	42	3	93,3	6,7	42	3	93,3	6,7	93,3
CV	8	6	2	75,0	25,0	5	3	62,5	37,5	68,8
	1889	**1542**	**347**	**81,6**	**18,4**	**1561**	**328**	**82,6**	**17,4**	**82,1**

Quadro 21 - Resultado detalhado da validação de cada especialista.

APÊNDICE H - Scripts da criação da Base de Dados

Script Para Criação da Tabela InfoBlast

```
-- Script para criação da tabela InfoBlast
-- Finalidade: Armazenar endereços html recuperados das páginas BLAST
-- usadas pelo anotador (Note), melhor resultado Blast (Blast)
-- ou de um organismo semelhante (Org)
-- St_Organismo: determina se o organismo é o mesmo (S/N)

CREATE TABLE InfoBlast (
        Id_ORF          numeric(11) NOT NULL default '0',
        Vl_Evalnote     varchar(10) NOT NULL default '',
        Vl_Evalblast    varchar(10) NOT NULL default '',
        Lk_Note         text NOT NULL,
        Vl_NoteQuery    numeric (16,4) NOT NULL default '0.0000',
        Vl_NoteSubje    numeric (16,4) NOT NULL default '0.0000',
        Lk_Blast        text NOT NULL,
        Vl_BlastQuery   numeric (16,4) NOT NULL default '0.0000',
        Vl_BlastSubje   numeric (16,4) NOT NULL default '0.0000',
        Lk_Organismo    text NOT NULL,
        Vl_OrgQuery     numeric (16,4) NOT NULL default '0.0000',
        Vl_OrgSubje     numeric (16,4) NOT NULL default '0.0000',
        St_Organismo    char(1) default 'N',
        Dt_Gravacao     timestamp,
        PRIMARY KEY (Id_Orf),
        FOREIGN KEY (Id_ORF) REFERENCES ORF     );
```

Script Para Criação da Tabela InfoNcbi

```
-- Script para criação da tabela InfoNCBI
-- Finalidade: Armazenar informações recuperadas das páginas html do NCBI
-- e PubMed, acessadas pelos links da tabela InfoBlast
-- Tp_Link: Define a importância e tipo de link pesquisado, sendo:
--      1 – Link referente pelo campo NotePad, usado pelo anotador
--      2 – Link do melhor resultado Blast, usado na ausência de um
--          link definido pelo anotador
--      3 – Link referente ao resultado blast com mesmo organismo
--      4 – Link referente a todos os hits do blast para o gene

CREATE TABLE InfoNcbi (
        Id_Lancamento   SERIAL,
        Id_ORF          numeric(11) NOT NULL default '0',
        Tp_Link         numeric (1) NOT NULL,
        Ds_Titulo       text NOT NULL,
        Ds_TermoTit     text,
        Tx_Resumo       text,
        Ds_TermoRes     text,
        Cd_Evidencia    varchar(60),
        Dt_Gravacao     timestamp,
        CONSTRAINT pk_InfoNcbi PRIMARY KEY (Id_Lancamento),
```

```
CONSTRAINT fk_InfoNcbi_Orf FOREIGN KEY (Id_ORF) REFERENCES ORF(Id_ORF));
```

Script Para Criação da Tabela EvidenceLevel

```
CREATE TABLE EvidenceLevel (
  cd_evidencia          character(3) NOT NULL,
  ds_evidencia          character varying(80) NOT NULL,
  nv_importancia        integer,
  CONSTRAINT            pk_evidencelevel PRIMARY KEY (cd_evidencia)
) WITHOUT OIDS;
```

Script Para Criação da Tabela EvidenceTermo

```
CREATE TABLE EvidenceTermo (
  id_termo              serial NOT NULL,
  cd_evidencia          character(3) NOT NULL,
  ds_termo              character varying(200) NOT NULL,
  CONSTRAINT pk_evidencetermo PRIMARY KEY (id_termo),
  CONSTRAINT termo_evidencia FOREIGN KEY (cd_evidencia)
     REFERENCES evidencelevel (cd_evidencia) MATCH SIMPLE
     ON UPDATE CASCADE ON DELETE CASCADE
) WITHOUT OIDS;
```

Script Para Criação da Tabela EvidenceRegra

```
CREATE TABLE EvidenceRegra (
  id_regra      serial NOT NULL,
  id_termo      integer,
  ds_regra      character varying(30),
  tp_regra      character varying(2) NOT NULL,
  CONSTRAINT pk_evidenceregra PRIMARY KEY (id_regra),
  CONSTRAINT fk_regra_termo FOREIGN KEY (id_termo)
     REFERENCES evidencetermo (id_termo) MATCH SIMPLE
     ON UPDATE CASCADE ON DELETE CASCADE
) WITHOUT OIDS;
```

Script Para Criação da Função AnalisaRegras

```
CREATE OR REPLACE FUNCTION AnalisaRegras(v_id_termo integer, v_ds_termo text, v_tx_artigo text)
  RETURNS boolean AS
$BODY$
DECLARE
   v_retorno     boolean;
   cur_regra     CURSOR (key integer) IS SELECT tp_regra, ds_regra FROM evidenceregra WHERE id_termo =
key;
   v_ds_regra    EvidenceRegra.ds_regra%TYPE;
   v_tp_regra    EvidenceRegra.tp_regra%TYPE;
   v_vl_termopos integer; -- posicao do termo no titulo ou resumo
   v_vl_regrapos integer;

BEGIN
   v_retorno = true;
   v_vl_termopos = position(v_ds_termo in v_tx_artigo);
```

```
    -- Carregar cursor com todas as regras
    OPEN cur_regra(v_id_termo);
    -- Enquanto existir regra e nenhuma falhar
    LOOP
       FETCH cur_REGRA INTO v_tp_regra, v_ds_regra;
       EXIT WHEN NOT FOUND OR NOT v_retorno;

       -- Ler e interpretar tipo de regra
       v_vl_regrapos = position(v_ds_regra in v_tx_artigo);

       IF (v_tp_regra = '<' AND v_vl_regrapos >= v_vl_termopos) THEN
          v_retorno = false;
       END IF;
       IF (v_tp_regra = '>' AND v_vl_regrapos <= v_vl_termopos) THEN
          v_retorno = false;
       END IF;
       IF (v_tp_regra = '-' AND v_vl_regrapos > 0) THEN
          v_retorno = false;
       END IF;
       IF (v_tp_regra = '+' AND v_vl_regrapos = 0) THEN
          v_retorno = false;
       END IF;
       IF (v_tp_regra = '-<' AND v_vl_regrapos < v_vl_termopos) THEN
          v_retorno = false;
       END IF;
       IF (v_tp_regra = '->' AND v_vl_regrapos > v_vl_termopos) THEN
          v_retorno = false;
       END IF;

    END LOOP;
    CLOSE cur_regra;

    RETURN(v_retorno);

END;
$BODY$   LANGUAGE 'plpgsql' VOLATILE;
```

Script Para Criação da Função InfereEvidência

```
CREATE OR REPLACE FUNCTION InfereEvidencia(v_id_lancamento integer, v_ds_titulo text, v_tx_resumo
text)
  RETURNS text AS
$BODY$
DECLARE
   v_nv_importancia   EvidenceLevel.nv_importancia%TYPE;
   v_cd_evidencia     EvidenceLevel.cd_evidencia%TYPE;
   v_ds_termotit      EvidenceTermo.ds_termo%TYPE;
   v_ds_termores      Evidencetermo.ds_termo%TYPE;
   v_ds_texto       text;
BEGIN

   v_nv_importancia = 0;

   update InfoNcbi
     set ds_termotit   = '',
         ds_termores   = '',
         cd_evidencia  = ''
    where id_lancamento = v_id_lancamento;
```

```
  LOOP

    v_ds_termotit      = '';
    v_ds_termores      = '';

    -- Seleciona nivel de evidencia de maior importancia
    select nv_importancia, cd_evidencia
      into v_nv_importancia, v_cd_evidencia
      from evidencelevel
     where (nv_importancia) = (select min(nv_importancia)
                      from evidencelevel
                     where nv_importancia > v_nv_importancia);

    IF v_nv_importancia is NULL THEN
       EXIT;
    END IF;

     -- Recupera para o nivel de evidencia de maior importancia
    -- caso exista algum termo relacionado com o titulo
    select ET.ds_termo
      into v_ds_termotit
      from evidencetermo ET
     where ET.cd_evidencia = v_cd_evidencia
       and position(ET.ds_termo in lower(v_ds_titulo)) > 0;

    IF (v_ds_termotit IS NOT NULL) THEN
       update InfoNcbi
         set ds_termotit   = ds_termotit  || ':' || v_ds_termotit,
             cd_evidencia  = cd_evidencia || ':' || v_cd_evidencia
       where id_lancamento = v_id_lancamento;
    END IF;

    -- Recupera para o nivel de evidencia de maior importancia
    -- caso exista algum termo relacionado com o resumo
    select ET.ds_termo
      into v_ds_termores
      from evidencetermo ET
     where ET.cd_evidencia = v_cd_evidencia
       and position(ET.ds_termo in lower(v_tx_resumo)) > 0;

    IF (v_ds_termores IS NOT NULL) THEN
       update InfoNcbi
         set ds_termores   = ds_termores  || ':' || v_ds_termores,
             cd_evidencia  = cd_evidencia || ':' || v_cd_evidencia
       where id_lancamento = v_id_lancamento;
    END IF;

  END LOOP;

  RETURN('Lançamento: '||v_id_lancamento);

END;$BODY$  LANGUAGE 'plpgsql' VOLATILE;
```

APÊNDICE I - Código fonte do GO-SIEVe

Código Fonte da Classe InfoBlast

```
public class InfoBlast {
   private static int Id_Orf = -1;
   private static boolean Is_Valid = false;
   private static String Vl_Evalnote = null;
   private static String Vl_Evalblast = null;
   // Link com campo Note Pad
   private static String Lk_Note;
   private static double Vl_NoteQuery = 0.00;
   private static double Vl_NoteSubje = 0.00;
   // Link do Blast
   private static String Lk_Blast;
   private static double Vl_BlastQuery = 0.00;
   private static double Vl_BlastSubje = 0.00;
   // Link para organismo semelhante
   private static String Lk_Organismo;
   private static double Vl_OrgQuery = 0.00;
   private static double Vl_OrgSubje = 0.00;
   private static char St_Organismo = 'N';

   //Variaveis temporï¿½rias para armazenar link com o Blast
   private static String LinkBlast;
   private static ProcessaArquivo Arquivo = new ProcessaArquivo();
   private static InfoNcbi Ncbi = new InfoNcbi();

   public InfoBlast() {
      Mensagem.MsgGeral("Classe: InfoBlast");
   }

   public static void main(String[] args) {
      String textoBlast;
      try {
         while (Id_Orf != 0) {

            // Baixa arquivo e converte em TXT
            SelecionaOrf();
            MontaLinkBlast();
            textoBlast = Arquivo.DownloadArquivo(LinkBlast);

            // Prepara arquivo BLAST baixado para ser processado
            ProcessaBlast Blast = new ProcessaBlast(textoBlast);

            // Recupera Link e Valores do melhor Blast
            Lk_Blast = Blast.ExtraiBlastMelhor();
            Vl_BlastQuery = Blast.valorQuery;
            Vl_BlastSubje = Blast.valorSubject;

            // Recupera Link e Valores do Blast usado pelo anotador
            Lk_Note = Blast.ExtraiBlastNote(AnotadorDados());
```

```
            Vl_NoteQuery = Blast.valorQuery;
            Vl_NoteSubje = Blast.valorSubject;

            // Recupera Link e Valores do Blast correspondente a um organismo semelhante
            Lk_Organismo = Blast.ExtraiBlastOrganismo();
            Vl_OrgQuery = Blast.valorQuery;
            Vl_OrgSubje = Blast.valorSubject;

            SelecionaLinks();
            GravaBlast();

            // Executa chamada para processamento de cada link Blast existente
            if (Is_Valid) {
               Ncbi.ProcessaNcbi(Id_Orf, Lk_Blast, '1');
               Ncbi.ProcessaNcbi(Id_Orf, Lk_Note, '2');
               Ncbi.ProcessaNcbi(Id_Orf, Lk_Organismo, '3');
               Blast.ExtraiTodosBlasts(Id_Orf);
            }

            Conexao.Commit();
//             Runtime.getRuntime ().gc ();
            System.gc();
         }
         InfereEvidencia();
         System.exit(0);
      } catch (Exception e) {
         Conexao.Erro.Excecao(e,"InfoBlast: Main:");
      }
   }

   // Verifica na base de Dados do SABIï¿½ uma ï¿½nica orf que ainda nï¿½so tenha sido processada
   private static void SelecionaOrf() {
      String Vl_Valids = "X";
      Mensagem.MsgGeral("Selecionando Orf");

      Conexao.ExecConsulta("Select id_orf From orf " +
                           " except " +
                           "Select id_orf From InfoBlast LIMIT 1");
      try {
         while (Conexao.RetornoSql.next()) {
            Id_Orf = Conexao.RetornoSql.getInt("id_orf");
         }
      } catch (Exception e) {
         Conexao.Erro.Excecao(e, "InfoBlast: SelecionaOrf");
      }

      Conexao.ExecConsulta("Select valids From orf where id_orf = " + Id_Orf);
      try {
         while (Conexao.RetornoSql.next()) {
            Vl_Valids = Conexao.RetornoSql.getString("valids").toUpperCase();
         }
         if (Vl_Valids.contains("Y") || Vl_Valids.contains("C") || Vl_Valids.contains("U") || Vl_Valids.contains("N")) {
            Is_Valid = true;
         }
      } catch (Exception e) {
         Conexao.Erro.Excecao(e, "InfoBlast: SelecionaOrf");
         Is_Valid = false;
      }
   }

   // Verifica orf no SABIÁ e monta link para baixar arquivo Blast correspondente no LNCC
```

```
private static void MontaLinkBlast() {
   Mensagem.MsgGeral("Criando Link para arquivo Blast no LNCC");

   Conexao.ExecConsulta("select a.id_blast, a.link_file as link " +
                        "from blast a " +
                       "where a.program_type = 'blastp' " +
                        "and a.id_orf       = " + Id_Orf + " " +
                        "and not (a.link_file like '%.orgPatog.%' " +
                         "or  a.link_file like '%.rs.%' " +
                         "or  a.link_file like '%.Scoelicolor.%')");
   try {
      while (Conexao.RetornoSql.next()) {
         LinkBlast = "http://www.brgene.lncc.br/webbie/annotation/" + Conexao.RetornoSql.getString("link");
      }
   } catch (Exception e) {
      Conexao.Erro.Excecao(e, "InfoBlast: MontaLinkBlast:");
   }
}

// Verifica os links extraidos
// - Se link Blast e do anotador forem diferentes indica que anotador usou outro critï¿½rio que nï¿½o o Blasr
// - Se link = ao do organismo, significa que anotador (ou Blast) identificaram um organismo semelhante
private static void SelecionaLinks() {
   Mensagem.MsgGeral("Selecionando Links vÃ¡lidos");
   St_Organismo = 'N';
   if (!Lk_Organismo.isEmpty() && (Lk_Blast.equals(Lk_Organismo) || Lk_Note.equals(Lk_Organismo))) {
      St_Organismo = 'S';
   }
   if (Lk_Blast.equals(Lk_Note)) {
      Lk_Blast = "";
      Vl_BlastQuery = 0.0;
      Vl_BlastSubje = 0.0;
   }
}

// Selecionando valor e-value usado pelo anotador
private static String[] AnotadorDados() {
   int posicaoIni;
   int posicaoFim;
   int i;
   String[] notePad = {"", "", "", ""};
   // 0 = Campo NotePad
   // 1 = Evalue
   // 2 = Score
   // 3 = Nome Organismo
   //= "", evalue = "";
   Mensagem.MsgGeral("Selecionando campo Note Pad usado pelo anotador");
   try {
      Conexao.ExecConsulta("Select A.notepad, A.product " +
            " From Annotation A " +
            "Where (A.id_annotation) = (Select max(B.id_annotation) " +
                                       "From annotation B " +
                                      "Where B.id_orf = A.id_orf) " +
                                       "and A.id_orf = " + Id_Orf);
      while (Conexao.RetornoSql.next()) {
         notePad[0] = Conexao.RetornoSql.getString("notepad").toLowerCase();
         notePad[3] = Conexao.RetornoSql.getString("product").toLowerCase();
      }
   } catch (Exception e) {
      Conexao.Erro.Excecao(e, "InfoBlast: AnotadorDados: ");
   }
```

```
            if (!(notePad[0] == null || notePad[0].isEmpty())) {
               Mensagem.MsgGeral("Selecionando valor Score do campo Note Pad");
               posicaoIni = notePad[0].indexOf("core =", 1);
               posicaoFim = notePad[0].indexOf(" bits", posicaoIni);
               if (posicaoIni != -1 && posicaoFim != -1) {
                  notePad[2] = notePad[0].substring(posicaoIni + 6, posicaoFim).trim();
               }
               Mensagem.MsgGeral("Selecionando valor e-value do campo Note Pad");
               posicaoIni = notePad[0].indexOf("expect =", 1);
               if (posicaoIni == -1) {
                  posicaoIni = notePad[0].indexOf("pected =", 1);
               }
               posicaoFim = notePad[0].indexOf("identiti", posicaoIni);
               if (posicaoFim == -1) {
                  posicaoFim = posicaoIni + 8 + 8;
               }
               if (posicaoIni != -1) {
                  notePad[1] = notePad[0].substring(posicaoIni + 8, posicaoFim).trim();
               }
            }

            Mensagem.MsgGeral("Preparando o nome do produto na anotação");
            if (!(notePad[3] == null || notePad[3].isEmpty())) {
               String[] produtoAdjetivo = I;

               for (i = 0; i < produtoAdjetivo.length; i++) {
                  notePad[3] = notePad[3].replaceAll(produtoAdjetivo[i], "");
               }
               notePad[3] = notePad[3].trim();
            }
            return (notePad);
         }

         // Com resultados de todos os tï¿½tulos calcula/compara as evidencias
         private static void InfereEvidencia() {
            Mensagem.MsgGeral("Inferindo Evidência");
            return;
         }

         public static void GravaBlast() {
            Mensagem.MsgGeral("Registando informações no BD");
            Conexao.ExecConsulta("Insert into InfoBlast (id_orf, vl_evalnote, vl_evalblast, lk_note, vl_notequery,
vl_notesubje," +
                                 " lk_blast, vl_blastquery, vl_blastsubje," +
                                 " lk_organismo, vl_orgquery, vl_orgsubje, st_organismo) " +
                                 "values (" + Id_Orf + "," + Vl_Evalnote + "," + Vl_Evalblast + ",'" +
                                 Lk_Note + "'," + Vl_NoteQuery + "," + Vl_NoteSubje + ", '" + Lk_Blast + "', " +
                                 Vl_BlastQuery + ", " + Vl_BlastSubje + ", '" + Lk_Organismo + "',  " +
                                 Vl_OrgQuery + ", " + Vl_OrgSubje + ", '" + St_Organismo + "')");
         }
      }
```

Código Fonte da Classe Conexão

```
      import java.sql.*;

      public class Conexao {
         private static String Driver = null;
```

```
private static String Url = null;
private static String Usuario  = null;
private static String Senha = null;
private static Connection MinhaConexao = null;
public static  ResultSet RetornoSql;
public static Excecao Erro = new Excecao("");

public Conexao () {
   Mensagem.MsgGeral ("Classe: Conexao");
   try {
      if (MinhaConexao == null ) {
         Driver  = "org.postgresql.Driver";
         Url = "jdbc:postgresql://10.32.1.4:5432/SabiaAnnotation/public";
         Usuario = "SabiaAnnotation";
         Senha   = "********";

         Class.forName (Driver);
         MinhaConexao = DriverManager.getConnection (Url ,Usuario, Senha);
         MinhaConexao.setAutoCommit (false);
         Mensagem.MsgGeral ("Conexão estabelecida com o BD");
      }
   } catch (Exception e) {
      Erro.Excecao (e, "Conexao:");
   }
}

public static Connection Conectar ()  {
   try {
      if (MinhaConexao == null) {
         Class.forName (Driver);
         MinhaConexao = DriverManager.getConnection (Url ,Usuario, Senha);
         MinhaConexao.setAutoCommit (false);
         Mensagem.MsgGeral ("Conexão estabelecida com o BD");
         return (MinhaConexao);
      } else {
      }
   } catch (Exception e) {
      Erro.Excecao (e, "Conexao: Conectar: ");
   } return (null);
}

// Armazena resultado da cunsulta no objeto e tambï¿½m o retorna
public static void ExecConsulta (String sql)  {
   Conectar ();
   try {
      Statement  MyState = MinhaConexao.createStatement ();
      RetornoSql = MyState.executeQuery (sql);
   } catch (Exception e) {
      Erro.Excecao(e, "Conexao: ExecConsulta:");
   }
}

public static void Desconectar () {
   try {
      if (!(MinhaConexao == null)) {
         RetornoSql.close ();
         MinhaConexao.close ();
         Mensagem.MsgGeral ("Encerrada conexão com o BD");
      }
   } catch (Exception e) {
      Erro.Excecao (e, "Conexao: Desconectar:");
```

```
        }
    }

    // Grava as informaÃ§Ãµes definitivamente no BD
    public static void Commit(){
        try {
            MinhaConexao.commit ();
        } catch (SQLException e) {
            Erro.Excecao(e, "Conexao: Commit:");
        }
    }

    // Se as variï¿½veis para conexï¿½o do BD estiverem em branco
    private void ConfigBD () {
        if (Driver.isEmpty ()   || (Driver ==  null))
            System.out.println ("Informe o Drive do Banco: ");
        if (Url.isEmpty ()      || Url == null)
            System.out.println ("Informe a Url do Banco: ");
        if (Usuario.isEmpty ()  || Usuario  == null)
            System.out.println ("Informe o nome do usuário autorizado no banco: ");
        if (Senha.isEmpty ()    || Senha == null)
            System.out.println ("Informe a senha do usuário no banco: ");
    }
}
```

Código Fonte da Classe Excecao

```
public class Excecao extends Exception {

    private Exception Exc;
    private String Erro;
    private Class Cla = null;

    public Excecao (String msg) {
        super (msg);
    }

    public void Excecao (Exception e){
        Exc = e;
        MostraErro();
        CorrigeErro();
    }

    public void Excecao (Exception e, String s) {
        Exc = e;
        try {
            Cla = Class.forName (s);
        } catch (Exception er) {
            this.Excecao(er, "Excecao");
        }
        MostraErro();
        CorrigeErro();
    }

    // Exibe detalhes do erro
    private void MostraErro () {
        Mensagem.MsgGeral("    Classe: "+Exc.getClass ());
        Mensagem.MsgGeral("     Causa: "+Exc.getCause ());
        Mensagem.MsgGeral("    Origem: "+super.toString ());
```

```
        Mensagem.MsgGeral("      Erro: "+Exc.getMessage ());
        Mensagem.MsgGeral("Comentario: "+Exc.toString ());

        if (Cla != null) {
           Mensagem.MsgGeral (" Classe: "+Cla.getClass ());
           Mensagem.MsgGeral ("  Nome : "+Cla.getSimpleName ());
        }
     }

     // Aï¿½ï¿½es para corrigir o erro
     private void CorrigeErro (){

        if (Exc.toString().equalsIgnoreCase("java.lang.NullPointerException"))
           Mensagem.MsgGeral ("ParÃ¢metro NULL passado ou recebido de um mÃ©todo");
        if (Exc.toString().equalsIgnoreCase("ClassNotFoundException"))
           Mensagem.MsgGeral ("Classe: "+Cla.toString () +" nï¿½o exsite!");

        return;
     }
  }
```

Código Fonte da Classe InfoNcbi

```
public class InfoNcbi {

   private static int      Id_Orf;
   private static char     Tp_Link; // 1 Blast - 2 Note - 3 Organismo
   private static String   Ds_Titulo;
   private static String   Tx_Resumo;
   private static String   ncbiHtml, ncbiTexto;
   private static ProcessaArquivo  Arquivo  = new ProcessaArquivo();
   private static Conexao          Bd       = new Conexao();
   public static  Excecao          Erro     = new Excecao("");

   public InfoNcbi () {
      Mensagem.MsgGeral ("Classe: ProcessaNcbi");
   }

   public void ProcessaNcbi (int IdOrf, String urlNcbi, char TipoUrl) {
      if (urlNcbi.isEmpty ()) {
         Mensagem.MsgGeral ("Nome do arquivo está vazio, cancelando processamento");
         return;
      }
      Id_Orf     = IdOrf;
      Tp_Link    = TipoUrl;
      ncbiTexto = Arquivo.DownloadArquivo (urlNcbi);
      ExtraiTitulo();
   }

   private void ExtraiTitulo () {
      String urlResumo, textoMem, resultadoNcbi;
//     String tituloInvalido[] = {"direct submission","genome sequence","complete genome sequence of",
//                                "the genome sequence of","complete dna sequence of", "complete genomic sequence of",
//                                 "dna sequence of","complete dna sequence",""};
      int posicaoIni = 1, posicaoFim = 1, i;
//     boolean isValid;

      // Localizar tag do TÃtulo no NCBI
      while ((posicaoIni = ncbiTexto.indexOf ("TITLE",posicaoFim)) != -1) {
```

```
                posicaoFim      = ncbiTexto.indexOf ("JOURNAL", posicaoIni);
                resultadoNcbi   = ncbiTexto.substring (posicaoIni+6, posicaoFim);
                Ds_Titulo       = resultadoNcbi.replaceAll ("'","*").trim ();
                Tx_Resumo = "";

                if (!Ds_Titulo.toLowerCase ().contains("direct submission")) {
                   // Recuperar link para o resumo no NCBI
                   posicaoIni      = ncbiTexto.indexOf ("=pubmed&list_uids=",posicaoFim);
                   if (posicaoIni != -1) {
                      urlResumo                                                                                    =
"http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=pubmed&dopt=Abstract&query_hl=1&list_uids=";
                      posicaoFim      = ncbiTexto.indexOf (">", posicaoIni);                    // era posicaoFim
                      urlResumo       = urlResumo + ncbiTexto.substring (posicaoIni+18, posicaoFim);
                      Mensagem.MsgGeral ("Link para Resumo NCBI: "+urlResumo);

                      // Se achar tÃtulo vÃ¡lido, extrai resumo no arquivo ArqTexto
                      ExtraiResumo (urlResumo);
                   } else {
                      urlResumo = "";
                   }
                   GravaNcbi ();
                }
             }
          }

          private void ExtraiResumo (String urlResumo) {
             String arqResumo;
             int posicaoIni = 1, posicaoFim = 1;
             arqResumo = Arquivo.DownloadArquivo (urlResumo);
             try {

                while ( (posicaoIni = arqResumo.indexOf ("</b>",posicaoFim+1)) != -1){
                   posicaoFim = posicaoIni;
                }
                posicaoIni = posicaoFim;

                if ( (posicaoFim = arqResumo.indexOf ("Publication Types:", posicaoIni)-12) < 0 ) {
                   posicaoFim = arqResumo.indexOf ("/>PMID:", posicaoIni)-7;
                }
                if (posicaoFim > 0){
                   arqResumo = arqResumo.substring (posicaoIni, posicaoFim);
                   posicaoIni = 1;
                   while ( (posicaoIni = arqResumo.indexOf ("<br />",posicaoIni+1)) != -1){
                      posicaoFim = posicaoIni;
                   }
                   arqResumo = arqResumo.substring ((posicaoFim+6));
                   Tx_Resumo = arqResumo.replaceAll ("'","*");
                }
                Mensagem.MsgGeral ("Resumo NCBI: "+Tx_Resumo);
             } catch (Exception e) {
                Erro.Excecao (e, "InfoNcbi:");
                Mensagem.MsgGeral ("Resumo NCBI: InvÃ¡lido, nÃ£o foi possÃvel recuperar");
             }
          }

          private void GravaNcbi (){
             if (!(Ds_Titulo.isEmpty ())) {
                Bd.ExecConsulta ("Insert into infoncbi (id_orf, tp_link, ds_titulo, tx_resumo) " +
                         "values ("+Id_Orf+", '"+Tp_Link+"', '"+Ds_Titulo+"', '"+Tx_Resumo+"')");
             }
          }
```

```
}
```

Código Fonte da Classe Mensagem

```
public class Mensagem {

   public static boolean MsgStatus = true;
   public static boolean MsgDebug;
   public static boolean MsgRastro;

   public Mensagem (){}

   public Mensagem (String Msg) {
      MsgStatus = true;
      MsgDebug = false;
      MsgRastro = false;
      MsgGeral(Msg);
   }

   public static void MsgGeral (String Msg) {
      if (MsgStatus)
         MsgStatus(Msg);
      if (MsgDebug)
         MsgDebug(Msg);
      if (MsgRastro)
         MsgRastro(Msg);
   }

   // Usada para depurar erros do programa
   private static void MsgStatus (String Msg) {
      System.out.println (Msg);
   }

// Usada para depurar erros do programa
   private static void MsgDebug (String Msg) {
      System.out.println (Msg);
   }

   // Usada para rastrear ações do usuário
   private static void MsgRastro (String Msg) {
      System.out.println (Msg);
   }
}
```

Código Fonte da Classe ProcessaArquivo

```
import java.io.*;
import java.net.*;

public class ProcessaArquivo {

   private static Excecao Erro = new Excecao("");
   public ProcessaArquivo () {
      Mensagem.MsgGeral ("Classe: Processa Arquivo");
   }

   // Recebe um link Origem e baixa arquivo da internet
```

```
public String DownloadArquivo (String urlString) {
   String textoMem;
   StringBuffer textoString = new StringBuffer();
   Mensagem.MsgGeral ("Download arquivo: "+urlString);
   try {
      if (urlString == null ){
         Mensagem.MsgGeral ("Url inválida (NULL)");
         return ("Url Invalida!!!!");
      }
      URL urlOrigem = new URL(urlString);
      HttpURLConnection htconn = (HttpURLConnection) urlOrigem.openConnection();
      htconn.setRequestMethod("GET");
      htconn.connect();
      InputStream in = htconn.getInputStream();
      InputStream buffer = new BufferedInputStream(in);
      Reader stream = new InputStreamReader(buffer);
      BufferedReader textoBuffer =  new BufferedReader(stream);

      while ((textoMem = textoBuffer.readChar ()) != null) {
         textoString.append(textoMem);
      }
      htconn.disconnect();
      Thread.sleep (3000);
      return (textoString.toString());

   } catch (Exception e) {
      Erro.Excecao (e, "ProcessaArquivo: DownloadArquivo:");
      return null;
   }
}


public String DownloadArquivo (String UrlOrigem, String UrlCfg) {
   String comando = "";
   try {
      if (UrlOrigem == null || UrlOrigem.equalsIgnoreCase ("C:\\Java\\"+UrlCfg+".html")){
         Mensagem.MsgGeral ("Url para Download inexistente: "+UrlOrigem);
      } else {
         UrlOrigem = UrlOrigem.trim ();
         UrlCfg    = UrlCfg.trim ();
         if (!UrlCfg.contains("Blast") && !UrlCfg.contains("Ncbi") && !UrlCfg.contains("Resumo") ) {
            Mensagem.MsgGeral ("Arquivo de configuraÃ§Ã£oo invÃ¡lido: "+UrlCfg);
         }
         comando = "C:\\Java\\Executaveis\\URL2File.Exe "+UrlOrigem+ " C:\\Java\\"+UrlCfg+".html";
         Mensagem.MsgGeral ("Executando: "+comando);
         Runtime.getRuntime().exec(comando);
         Thread.sleep (10000);
      }
      return ("C:\\Java\\"+UrlCfg+".html");
   } catch (Exception e) {
      Erro.Excecao (e, "ProcessaArquivo: DownloadArquivo:");
   }
   return (null);
}

// Converte o arquivo HTML em TXT
// Usa ferrameta de terceiro para covnerter - HtmlasText
private void HttptoTxt (String UrlCfg) {
   Mensagem.MsgGeral ("Convertendo arquivo "+UrlCfg+" .html para "+UrlCfg+".txt");
   String comando = "";
   try {
```

```
            // Execute a command without arguments
            comando = "C:\\Java\\Executï¿½veis\\htmlastext.exe /run C:\\Java\\"+UrlCfg+".cfg";
            Mensagem.MsgGeral ("Executando: "+comando);
            Process child = Runtime.getRuntime().exec(comando);
         } catch (Exception e) {
            Erro.Excecao (e, "ProcessaArquivo: HttpToTxt:");
         }
      }

      // Prepara arquivo para ser trabalhado
      public String PreparaArquivo (String arquivoHtml){
         String textoMem;
         FileInputStream htmlFile;
         InputStreamReader htmlStream;
         BufferedReader textoBuffer;
         StringBuffer textoString = new StringBuffer();

         try {
            htmlFile    =  new FileInputStream(arquivoHtml);
            htmlStream  =  new InputStreamReader(htmlFile);
            textoBuffer =  new BufferedReader(htmlStream);
            while ((textoMem = textoBuffer.readLine ()) != null) {
               textoString.append (textoMem);
            }
            htmlFile.close ();
            new File(arquivoHtml).delete ();

            return (textoString.toString ());
         } catch (Exception e) {
            Erro.Excecao (e, "ProcessaArquivo: PreparaArquivo:");
         }
         return("");
      }
   }
```

Código Fonte da Classe ProcessaArquivo

```
   public class ProcessaBlast {
      private String arquivoHtml;
      private String textoBlast;
      public int valorQuery = 0;
      public int valorSubject = 0;
      private String evalueBlast = "";
      private String scoreBlast = "";
      private String organismoBlast = "";
      private String url = "";

      private String melhorUrl;
      private int melhorQuery;
      private int melhorSubject;
      public static Excecao Erro = new Excecao("");

       public ProcessaBlast () {
         Mensagem.MsgGeral ("Classe: ProcessaBlast");
      }

      public ProcessaBlast (String texto) {
         Mensagem.MsgGeral ("Classe: ProcessaBlast");
         textoBlast = texto;
```

```
    }

    // Extrai link para NCBI referente ao melhor resultado Blast
    public String ExtraiBlastMelhor () {
       int posicaoIni, posicaoFim, x;
       String resultadoBlast;
       Mensagem.MsgGeral ("Extraindo link do melhor Blast");
       String Url[] = {"<a href=\"http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids=",
                       "<a href=\"http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids="};

       for (x=0; x<2; x++) {
          // Localizar tag do melhor resultado Blast
          // Recuperar Linha inteira
          posicaoIni = textoBlast.indexOf (Url[x],1);
          posicaoFim = textoBlast.indexOf (Url[x], posicaoIni+Url[x].length ());
          if (posicaoIni > 0 && posicaoFim > 0) {
             resultadoBlast = textoBlast.substring (posicaoIni, posicaoFim);
             Mensagem.MsgGeral (resultadoBlast);
             // Extrair link, e-value e referencias para
             SeparaBlast(resultadoBlast);
             return (this.url);
          }
       }
       return ("");
    }

    // Extrai link para NCBI referente ao resultado Blast utilizado no campo Note Pad
    public String ExtraiBlastNote (String[] dadosSabia) {
       int posicaoIni = 1, posicaoFim = 1, x, y = 1;
       String resultadoBlast = "";
        this.melhorQuery     = 0;          this.valorQuery     = 0;
       this.melhorSubject   = 0;          this.valorSubject   = 0;
       this.melhorUrl       = "";

        try {
          posicaoIni = dadosSabia[0].indexOf("gi|");
          if (posicaoIni != -1) {
             posicaoFim = dadosSabia[0].indexOf("|", posicaoIni+4);
             if (posicaoIni != -1 && posicaoFim != -1) {
                this.melhorUrl = "http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids=";
                this.melhorUrl       =       this.melhorUrl       +       (dadosSabia[0].substring(posicaoIni       +       3,
posicaoFim).trim())+"&dopt=GenPept";
             }
          } else {
             String                                tag[]                                =                                {"<a
href=\"http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids=",
                          "<a
href=\"http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids="};
             for (x=0; x<2; x++) {
                // Localizar tag do resultado Blast usado pelo anotador
                // Recuperar Linha inteira
                while (!dadosSabia[3].isEmpty () && this.evalueBlast.length () < 6 && y <= 3) {
                   posicaoIni = textoBlast.indexOf (tag[x],posicaoFim);
                   posicaoFim = textoBlast.indexOf (tag[x], posicaoIni+tag[x].length ());
                   resultadoBlast = textoBlast.substring (posicaoIni, posicaoFim);
                   Mensagem.MsgGeral (resultadoBlast);

                   // Extrair link, e-value e referencias para o resultado Blast do campo NotePad
                   // Pesquisa todo arquivo Blast
                   SeparaBlast(resultadoBlast);
                   y = this.AvaliaBlast(dadosSabia, y);
```

```
            }
          }
        }
    } catch (Exception e) {
        Erro.Excecao(e, "ProcessaBlast: ExtraiBlastNote:");
    }
    this.url           = this.melhorUrl;
    this.valorQuery    = this.melhorQuery;
    this.valorSubject  = this.melhorSubject;
    return (this.url);
}

// Extrai link para NCBI referente ao resultado Blast correspondente a um organismo semelhante
public String ExtraiBlastOrganismo () {
    Mensagem.MsgGeral ("Extraindo link Blast mesmo organismo");
    int posicaoIni = -1, posicaoFim = 1, x;
    String resultadoBlast = "a";

    String urlNcbi = "";
    String tag[] = {"<a href=\"http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids=",
                    "<a href=\"http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids="};

    try {
        for (x=0; x<2; x++) {

            // Localizar tag do resultado Blast de um mesmo organismo
            // Recuperar Linha inteira
            while (evalueBlast.length () < 6 && posicaoIni == -1) {
                posicaoIni = textoBlast.indexOf (tag[x],posicaoFim);
                posicaoFim = textoBlast.indexOf (tag[x], posicaoIni+tag[x].length ());
                resultadoBlast = textoBlast.substring (posicaoIni, posicaoFim);
                Mensagem.MsgGeral (resultadoBlast);

                // Extrair link, e-value e referencias para
                SeparaBlast(resultadoBlast);

                // Verifica se existe no link o mesmo nome de organismo (violaceum)
                posicaoIni = this.organismoBlast.indexOf ("violaceum");
            }
            if (posicaoIni != -1) {
                Mensagem.MsgGeral ("Link organismo igual encontrado: "+urlNcbi);
                return(this.url);
            }
        }
    } catch (Exception e) {
        Erro.Excecao(e, "ProcessaBlast: ExtraiBlastOrganismo: ");
    }
    return ("");
}

private void SeparaBlast (String linha) {
    String urlNcbi, organismo, href, evalue, length, score, identidade, positivos, gaps, query;
    int posicaoIni, posicaoFim, i;

    this.valorQuery    = 0;
    this.valorSubject  = 0;
    this.evalueBlast   = "";
    this.organismoBlast = "";
    this.scoreBlast    = "";

    // Extrai referencia para dados de query e subject
```

```
if ((posicaoIni = linha.indexOf ("http://www.ncbi.nlm.nih.gov")) == -1) {
     this.evalueBlast = "xxxxxxxx";
     return;
} else {
    posicaoFim = linha.indexOf ("\"", posicaoIni);
    this.url = linha.substring (posicaoIni, posicaoFim);

    // Nome do organismo
    posicaoIni = linha.indexOf ("</a>", posicaoFim)+44;
    posicaoFim = linha.indexOf ("<a href = #", posicaoIni);
    organismo = linha.substring (posicaoIni, posicaoFim);
    organismo = this.NomeOrganismo(organismo);
    Mensagem.MsgGeral ("Organismo: "+organismo);
    this.organismoBlast = organismo;

    // Verifica referencia para detalhes do arquivo
    posicaoIni = linha.indexOf ("<a href = #", posicaoIni)+111;
    posicaoFim = linha.indexOf (">", posicaoIni);
    href = linha.substring (posicaoIni, posicaoFim).trim();


    // Extrai evalue
    posicaoIni = linha.indexOf ("</a>", posicaoIni)+44;
    evalue = linha.substring (posicaoIni).trim ();
    this.evalueBlast = evalue;

    // Extrai Score
    score = linha.substring (posicaoFim+1, posicaoIni-4).trim ();
    this.scoreBlast     = score;

    // Localiza Tag para detalhes da comparacao genetica
    posicaoIni = textoBlast.indexOf ("<a name = "+href);
    posicaoFim = textoBlast.indexOf ("Query:", posicaoIni)-5;
    if (posicaoIni > 0 && posicaoFim > 0 ) {
       linha = textoBlast.substring (posicaoIni, posicaoFim);

       // Nome do organismo
       posicaoIni = linha.indexOf ("</a>", 50)+44;
       posicaoFim = linha.indexOf ("Length =",posicaoIni);
       organismo = linha.substring (posicaoIni, posicaoFim);
       organismo = this.NomeOrganismo(organismo);
       this.organismoBlast = organismo;

       // Extrai tamanho da sequencia
       posicaoIni = linha.indexOf ("Length =",posicaoIni)+81;
       posicaoFim = linha.indexOf ("Score", posicaoIni)-1;
       length = linha.substring (posicaoIni, posicaoFim).trim ();
       this.valorSubject   = Integer.parseInt (length);

       // Extrai score
       posicaoIni = linha.indexOf ("Score ", posicaoIni)+77;
       posicaoFim = linha.indexOf (" bits", posicaoIni);
       score = linha.substring (posicaoIni, posicaoFim).trim ();
       this.scoreBlast     = score;

       // Extrai identidade
       posicaoIni = linha.indexOf ("Identities =", posicaoIni)+122;
       posicaoFim = linha.indexOf ("/", posicaoIni);
       identidade = linha.substring (posicaoIni, posicaoFim).trim ();

       // Extrai Query
```

```
            query = linha.substring (posicaoFim+1, linha.indexOf ("(",posicaoFim)).trim();
            this.valorQuery     = Integer.parseInt (query);

            // Extrai positivos
            posicaoIni = linha.indexOf ("Positives =", posicaoIni)+111;
            posicaoFim = linha.indexOf ("/", posicaoIni);
            positivos = linha.substring (posicaoIni, posicaoFim).trim ();

            // Extrai gaps
            posicaoIni = linha.indexOf ("Gaps =", posicaoIni);
            posicaoFim = linha.indexOf ("/", posicaoIni);
            gaps = "0";
            if (posicaoIni > 0 && posicaoFim > 0) {
               gaps = linha.substring (posicaoIni+6, posicaoFim).trim ();
            }
         }
      }
   }

   private int AvaliaBlast (String[] dadosSabia, int y) {
      // testa se score Ã© igual
      if (y == 1 && dadosSabia[2].equals (this.scoreBlast)) {
         this.melhorUrl      = this.url;
         this.melhorQuery    = this.valorQuery;
         this.melhorSubject  = this.valorSubject;
         y++;
      }
      // testa se evalue Ã© igual
      if (y == 1 && !this.evalueBlast.isEmpty() && dadosSabia[1].contains(this.evalueBlast)) {
         this.melhorUrl      = this.url;
         this.melhorQuery    = this.valorQuery;
         this.melhorSubject  = this.valorSubject;
         y++;
      }
      // testa se organismo Ã© igual
      if (y == 1 && (this.organismoBlast.contains (dadosSabia[3]))) {
         this.melhorUrl      = this.url;
         this.melhorQuery    = this.valorQuery;
         this.melhorSubject  = this.valorSubject;
         y++;
      }
      // testa se score e evalue sÃ£o iguais
      if (y == 2 && dadosSabia[1].contains(this.evalueBlast) && dadosSabia[2].equals (this.scoreBlast)) {
         this.melhorUrl      = this.url;
         this.melhorQuery    = this.valorQuery;
         this.melhorSubject  = this.valorSubject;
         y++;
      }
      // testa se organismo e score sÃ£o iguais
      if (y == 2 && dadosSabia[2].equals (this.scoreBlast) && (this.organismoBlast.contains (dadosSabia[3]))) {
         this.melhorUrl      = this.url;
         this.melhorQuery    = this.valorQuery;
         this.melhorSubject  = this.valorSubject;
         y++;
      }
      // testa se organismo e evalue sÃ£o iguais
      if (y == 2 && dadosSabia[1].contains(this.evalueBlast) && (this.organismoBlast.contains(dadosSabia[3]))) {
         this.melhorUrl      = this.url;
         this.melhorQuery    = this.valorQuery;
         this.melhorSubject  = this.valorSubject;
         y++;
```

```
                }
                // testa se evalue, score e organismo são iguais
                if (y == 3 && dadosSabia[1].equals (this.evalueBlast) && dadosSabia[2].equals (this.scoreBlast) &&
(this.organismoBlast.contains (dadosSabia[3]))) {
                    this.melhorUrl      = this.url;
                    this.melhorQuery    = this.valorQuery;
                    this.melhorSubject  = this.valorSubject;
                    y++;
                }
                return (y);
            }

            private String NomeOrganismo(String organismo) {
                int i;
                if (!(organismo == null || organismo.isEmpty ())) {
                    organismo = organismo.trim ().toLowerCase ();
                    String produtoAdjetivo[] = {"conserved", "hypothetical", "protein", "putative", "probable", ",", "'", "\""};
                    for (i = 0; i < produtoAdjetivo.length; i++) {
                        organismo = organismo.replaceAll (produtoAdjetivo[i],"");
                    }
                }
                return (organismo);
            }

            public void ExtraiTodosBlasts (int Id_Orf) {
                int posicaoIni = 1, posicaoFim = 1, x;
                String resultadoBlast = "";
                InfoNcbi Ncbi = new InfoNcbi();
                String tag[] = {"<a href=\"http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids=",
                                "<a href=\"http://www.ncbi.nlm.nih.gov:80/entrez/query.fcgi?cmd=Retrieve&db=Protein&list_uids="};

                Mensagem.MsgGeral ("Extraindo todos os links Blast");
                for (x=0; x<2; x++) {
                    // Localizar tag do resultado Blast de um mesmo organismo
                    // Recuperar Linha inteira
                    try {
                        while (posicaoIni > 0) {
                            posicaoIni = textoBlast.indexOf (tag[x],posicaoFim);
                            posicaoFim = textoBlast.indexOf (tag[x], posicaoIni+tag[x].length ());
                            if (posicaoIni > 0) {
                                resultadoBlast = textoBlast.substring (posicaoIni, posicaoFim);
                                // Extrair link, e-value e referencias para
                                SeparaBlast(resultadoBlast);
                                Ncbi.ProcessaNcbi(Id_Orf, this.url, '4');
                            }
                        }
                    } catch (Exception e) {
                        Erro.Excecao(e, "ProcessaBlast: ExtraiTodosBlasts: Link NCBI nÃ£o encontrado ");
                    }
                }
            }
        }
```

ANEXOS

Anexo 1 - Exemplo de *download* de um resultado do blast (CV00068)

Anexo 2 - Exemplo de *download* de uma página no NCBI referenciada pelo Blast (CV00068)