2. Preliminares

2.1 Introdução

Este capítulo possui uma discussão preliminar sucinta dos contextos biológico e computacional, necessária para o entendimento e motivação deste trabalho.

Na discussão do contexto biológico serão apresentados os conceitos principais relacionados aos projetos genoma e proteoma. Uma visão mais detalhada sobre este assunto pode ser encontrada em [Casey, 2004; HGP, 2004a, b; Lemos, 2000a].

Na discussão do contexto computacional, será dada uma perspectiva geral de Bioinformática. Brevemente, a *Bioinformática* é uma área da informática que auxilia os pesquisadores em Biologia a criar, melhorar, desenvolver e manipular os bancos de dados de Biologia Molecular e outras ferramentas computacionais para coletar, organizar e interpretar os dados experimentais obtidos em laboratórios [HGP,2004a; Kanehisa, 1998; Kim, 2002].

Nesta discussão será dada uma visão geral dos assuntos tratados em Bioinformática relevantes à esta tese, como os programas de análise, os bancos de dados, os sistemas de anotação, a integração dos dados e aplicativos, as ontologias, e os workflows. O capítulo termina com uma discussão geral do sistema de gerência de análises em biossequências que será proposto nesta tese, e com um levantamento dos principais trabalhos relacionados.

2.2 Contexto Biológico

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

- mapear e sequenciar o genoma humano inteiro, obtendo as 3 bilhões de bases da cadeia que representam o DNA humano;
- identificar os aproximadamente 30.000 genes no DNA humano;
- armazenar esta informação em bancos de dados;
- melhorar as ferramentas de análises dos dados;
- transferir as tecnologias relacionadas ao setor privado; e
- tratar das consequências éticas, legais e sociais que surgirem com o projeto.

Cada célula de um organismo vivo contém cromossomos, que são compostos de uma sequência de pares de bases de DNA. O conjunto dessas sequências formam o genoma, código que traz instruções para controle da replicação e do funcionamento do organismo.

A saber, o DNA é uma sequência linear de quatro nucleotídeos (também chamados de bases, representados pelos caracteres *A*, *T*, *C* e *G*), que é a fonte básica da informação genética. Esta informação é copiada ou transcrita para moléculas de RNA, cujas sequências de nucleotídeos contém o código para a ordenação específica de uma sequência de aminoácidos (também chamados de resíduos, representados por 20 caracteres). As proteínas, que são sequências de aminoácidos, são então sintetizadas num processo que envolve a tradução do RNA.

Além do genoma humano, outros organismos estão sendo sequenciados. Com isso, novas tecnologias de mapeamento, sequenciamento e análise de *biossequências* (ou, simplesmente, sequências) estão sendo desenvolvidas, várias informações biológicas que trarão avanços em diversos campos, como agricultura, Biologia e Medicina, estão sendo descobertas, e melhorias nas análises do genoma humano estão sendo alcançadas. As sequências destes organismos facilitam a elucidação de funções de genes e sequências do genoma humano, pois há um princípio biológico que diz que se duas sequências, sejam nucleotídicas ou protéicas, são similares, então é razoável supor que suas funções também sejam similares.

Alguns objetivos do projeto genoma são [Sousa, 2001]:

- Compreensão mais ampla da organização do genoma e da função dos genes humanos, pela comparação com outros genomas já sequenciados;

- Diagnóstico antecipado de doenças;
- Desenvolvimento de drogas específicas para certos indivíduos;
- Cura de doenças genéticas;
- Determinação da predisposição genética de indivíduos a doenças como o câncer.

Diferente do projeto genoma humano, nem todos os projetos têm o objetivo de sequenciar o genoma inteiro do organismo. Muitos projetos estão focados em obter apenas os genes expressos, chamados de ESTs, o que simplifica muito a tarefa e consequentemente, reduz os custos.

EST (*expressed sequence tag*, em inglês) é uma sequência obtida de forma aleatória, geralmente incompleta, de DNA, que representa um gene expresso e que pode ser utilizado para identificar (*tag*) o gene. Um gene expresso é aquele cujo produto, seja uma proteína ou um RNA, está sendo produzido em um dado momento em uma célula.

Estes projetos tiram vantagem do fato de ser relativamente simples fazer cópias de DNAs a partir de mRNAs durante o processo de tradução. Estas cópias, chamadas de cDNAs, quando sequenciadas, são nomeadas de ESTs.

Considerando que os custos dos projetos genoma são elevados e que a quantidade de sequências não codificantes em eucariotos é muito maior que em procariotos, os projetos genoma do tipo ESTs são mais comuns para eucariotos[1], e os projetos completos para procariotos. Para se ter uma idéia, em um genoma de eucariotos, 90% do DNA não codifica para proteínas, tornando a obtenção das sequências dos genes que as codificam muito mais direta.

Apesar dos genomas variarem de tamanho de milhões de nucleotídeos, em uma bactéria, para bilhões de nucleotídeos, em humanos e em grande parte dos animais e plantas, as reações químicas que os pesquisadores utilizam para decodificar os pares de bases do DNA durante o processo de sequenciamento são precisas para se obter apenas 600 a 700 nucleotídeos por vez. Sendo assim, um dos processos de sequenciamento mais utilizados, conhecido como *shotgun*, se inicia com a quebra do DNA em milhões de fragmentos aleatórios, que são então

[1] De forma simplificada, um organismo procariótico é aquele cujo material genético não está organizado dentro do núcleo; ao contrário do organismo eucariótico, que possui núcleo e outras diferenças menores.

"lidos" por uma máquina de sequenciamento de DNA. A análise computacional é então usada para construir cromatogramas (consistindo de quatro curvas de cores diferentes, cada curva representando um sinal para uma das quatro bases), e para converter cada cromatograma em uma sequência de bases (chamadas de *reads*) por um programa específico. Exemplos de programas que realizam essa tarefa tem-se o Phred [Ewing, 1998a,b], o Abiview [Klatte, 2004] e o Chromas [Technelysium, 2004].

Em seguida, um programa chamado *assembler*, ou montador (por exemplo, os programas CAP3 [Huang, 1999] e Phrap [Green, 2004]), é incumbido da tarefa de combinar os *reads* para reconstruir a sequência original [Lemos, 2003a; Pop, 2002]. Infelizmente, este processo não é simples. Os dados podem conter erros, sejam por causa de limitações na tecnologia de sequenciamento ou por falha humana durante o trabalho de laboratório, e mesmo na ausência de erros há características das sequências de DNA que complicam esse trabalho (um exemplo são as regiões repetitivas, chamadas de repetições).

Na prática, a cobertura imperfeita, repetições e erros de sequenciamento fazem com que a montagem una apenas partes dos *reads*. Cada conjunto de *reads* que se une é chamado de contig. O objetivo final é unir todos os contigs gerados e formar um único contig, que representará um cromossomo completo. A tarefa de fechar os buracos entre os contigs e obter a molécula completa é chamada de *finishing*.

Como projetos de genoma geram dados brutos, ou seja, sem significado biológico, há outra fase, chamada de fase de anotação, cujo objetivo consiste em converter esses dados em informações biologicamente relevantes (sequências anotadas).

Uma anotação é uma meta-informação ou uma descrição de características em mais alto nível da sequência biológica, ou biossequência. Anotações úteis incluem vários tipos de informações, por exemplo, se um trecho de DNA contém um gene e qual a função dele.

Os pesquisadores usam diversas ferramentas e programas de computador, combinados com interpretação humana, para realizar esse processo de anotação. Porém, considerando a velocidade com que pesquisadores geram sequências de DNA atualmente, anotar os dados automaticamente torna-se um desafio

computacional, e a análise humana cuidadosa está se tornando cada vez mais difícil.

Apesar do sequenciamento completo do genoma humano ser um feito marcante na história da ciência, ele abre inúmeras possibilidades para novos desafios científicos de igual ou maior magnitude. Entre eles, destaca-se o projeto proteoma.

Proteoma significa o conjunto de proteínas expressas a partir de um determinado genoma. As proteínas são muito importantes pois exercem papel fundamental em todos os fenômenos biológicos. Apesar da sequência de aminoácidos de uma proteína ser definida pelas informações contidas no DNA, não é possível deduzir o proteoma conhecendo-se o genoma.

Enquanto o genoma de um organismo é praticamente constante e independente do tipo celular analisado, cada tipo de célula possui apenas parte do total de genes do genoma apta para formar proteínas, ou seja, ela expressa somente um subconjunto de seus genes. Portanto, cada célula possui seu proteoma específico. Além disso, após a tradução, as proteínas podem sofrer modificações químicas que não estavam codificadas no genoma, fazendo com que a variedade de proteínas aumente. Desta forma, o projeto proteoma ajudará a compreender porquê células com genomas iguais desempenham funções diferentes.

2.3 Programas de Análise

Ao isolar novas sequências moleculares em laboratório, os pesquisadores querem saber o máximo possível sobre essas sequências. O primeiro passo para isso é verificar se outros pesquisadores já estudaram sequências similares a essas. Provavelmente a ferramenta computacional mais usada em Biologia é o BLAST - *Basic Local Alignment Tool* [Altschul, 1998; Altschul, 1990; WU-BLAST, 2004a; Lemos, 2000a, b; Casanova, 2001; Lemos, 2003b] – que procura em bancos de dados, como o Genbank [Benson et al., 2003; NCBI, 2004a] todas as sequências similares a uma determinada sequência.

Os alinhamentos oferecem um ótimo meio de comparar sequências relacionadas. Um alinhamento pode ser global ou local, dependendo do propósito da comparação. Os alinhamentos globais forçam um alinhamento completo das

sequências de entrada, enquanto o local apenas detecta seus segmentos mais similares.

O primeiro algoritmo projetado para detectar alinhamento global ótimo foi o de Needleman-Wunsch [Needleman, 1970]. Depois, uma pequena variante foi proposta, chamada de algoritmo de Smith-Waterman [Smith, 1981], que encontra o alinhamento local ótimo entre duas sequências. Estes dois algoritmos requerem tempo de execução proporcional ao produto dos comprimentos das duas sequências que estão sendo comparadas.

Como as similaridades entre as sequências de proteínas e DNA frequentemente se localizam em apenas segmentos das sequências envolvidas, os programas de busca de similaridades mais populares são os baseados no algoritmo de Smith-Waterman. No entanto, devido à complexidade desses algoritmos, eles se tornam muito lentos para a maior parte dos usuários, que não dispõe dos recursos necessários para utilizá-los. Assim, os programas FASTA [Pearson, 1998] e BLAST usam heurísticas para concentrar seus esforços nas regiões das sequências com maiores probabilidades de estarem relacionadas.

A comparação de sequências entre pares de sequências pode ser generalizada para múltiplas sequências. CLUSTAL W [Thompson, 1994] e o MultAlign [CBRG, 2004] são exemplos de algoritmos para alinhamento múltiplo de sequências [Corpet, 1988].

Além da comparação de sequências, ferramentas são necessárias para a predição de genes, comparação de genomas, predição de estruturas, análise filogenética, descoberta e reconhecimento de padrões, entre outros. GLIMMER [Delcher et al., 1999] e ORFFinder [NCBI, 2004f] são exemplos de programas de predição de genes. TRIPOS [SYBYL, 2004], Modeller [Sali, 2004] e Threader [Jones, 2004] são programas de predição de estruturas de proteínas. PHYLIP [Felsenstein, 2004] (*the PHYLogeny Inference Package*) e PAUP [Swofford, 2004] (*Phylogenetic Analysis Using Parsimony*) são os pacotes de análise filogenética mais populares. Existem ainda o GCG [MRC, 2004] e o EMBOSS [Rice, 2000], pacotes com vários programas para a análise de sequências.

Para permitir uma comparação direta de sequências genômicas de organismos suficientemente similares, existem ferramentas de software que podem alinhar mais de duas sequências genômicas. Alfresco [Sanger, 2004] e

MGA (*Multiple Genome Aligner*) [Höhl, 2002] são exemplos de programas projetados para comparação de vários genomas.

Sequências de nucleotídeos e aminoácidos contêm padrões que foram preservados durante a evolução, pelo fato de serem importantes para a estrutura ou funcionamento da molécula. Em proteínas, essas sequências conservadas podem estar envolvidas na ligação ao seu substrato ou a outra proteína, podem ser o sítio ativo de uma enzima ou podem determinar a estrutura tridimensional da proteína. Sequências de nucleotídeos fora de regiões codificantes em geral tendem a ser menos conservadas entre organismos, exceto quando estão envolvidas no processo de regulação da expressão gênica. A descoberta de padrões em sequências de proteínas e nucleotídeos pode levar à determinação de função e descoberta de relacionamentos evolutivos entre sequências. Exemplos de algoritmos de descoberta de padrões são TEIRESIAS [Rigoutsos,1998], Pratt [Jonassen, 1996] e Blocks Maker [Henikoff et al., 1995].

Na descoberta de padrões [Brejová et al., 2004; Lemos, 2003c], o algoritmo deve previamente encontrar padrões desconhecidos nas sequências, sem que se conheça sua função ou importância. Porém, na Biologia, muitas sequências consenso são conhecidas. Assim, é importante ter ferramentas que procuram ocorrências de padrões conhecidos em novas sequências. Esse problema chama-se casamento de padrões. O RBSFinder [TIGR, 2004a] é um exemplo de programa de casamento de padrões que encontra sítios de ligação ribossomal.

2.4 Bancos de Dados

O número de bancos de dados de Biologia Molecular está crescendo rapidamente. Alguns bancos de dados se concentram em moléculas ou funções específicas e oferecem informações detalhadas, enquanto outros tentam cobrir uma área mais ampla da Biologia, com informações menos detalhadas.

Em alguns casos, as informações biológicas são geradas por análise computacional de outros bancos; em outros, é obtida da literatura ou de informações definidas pelos pesquisadores.

O Genbank-NT [NCBI, 2004c], o GSDB [Harger et al., 1998], o GDB [Letovsky et al., 2004] e o EMBL [Kulikova et al., 2004] são exemplos de banco de dados de sequências de nucleotídeos, já o Swiss-Prot [Boeckmann, 2003], o

PIR [Wu et al., 2002] e o Genbank-NR [NCBI, 2004c] são bancos de sequências de aminoácidos. O PDB (*Protein Data Bank*) [Westbrook et al., 2002] é um banco de dados de estruturas terciárias de sequências de aminoácidos.

O PROSITE [Falquet et al., 2002] é derivado do Swiss-Prot e armazena padrões de sequências protéicas conservados, associados a funções específicas. Existem outras bibliotecas de padrões, como Blocks [Henikoff et al., 2000] e PRINTS [Attwood et al., 2003], assim como bibliotecas que oferecem padrões de sequências de estruturas de domínios protéicos mais longos, como o PFam [Bateman et al., 2002] e o ProDom [Servant et al., 2002]. Uma das principais diferenças entre os bancos de dados de padrões é como os padrões são representados, se utilizando expressões regulares, alinhamentos múltiplos, ou matrizes de *Hidden Markov Model*.

2.5 Sistemas de Anotação

No processo de anotação, o desafio em Bioinformática está na criação de ferramentas efetivas para ajudar os pesquisadores a analisarem grandes conjuntos de biossequências através do acesso às anotações armazenadas em fontes de dados públicas; da execução de programas de análise e da criação de anotações manuais resultantes das análise feitas. Todas estas etapas devem contar com ajuda de uma interface apropriada.

Para tanto, os sistemas de anotações [Lemos, 2003d; Lemos, 2004b] devem ter modelados tanto as anotações externas, armazenadas em fontes de dados públicas externas, quanto as anotações internas, armazenadas no *data warehouse* sob controle do sistema, além de oferecer mecanismos de extensão do modelo para acomodar novas anotações.

As anotações internas podem ser classificadas como:

- Importadas: obtidas de fontes de dados públicas;
- Automáticas: geradas por programas de análise e
- Manuais: diretamente criadas pelo pesquisador.

Além disso, as características das anotações também variam de acordo com o tipo do projeto genoma. Anotações geradas no contexto de um projeto que objetiva a obtenção do DNA completo de um organismo têm requisitos diferentes daquelas criadas no contexto de um projeto cujo objetivo é obter ESTs.

Para modelar as anotações importadas é necessário entender os esquemas das fontes de dados externas.

Para modelar as anotações automáticas, é necessário entender as aplicações, suas entradas e saídas, o que, às vezes, é uma tarefa complicada pois nem sempre os programas possuem uma boa documentação e geralmente acontecem problemas durante suas execuções.

Para as anotações manuais, o sistema pode ter um vocabulário controlado, o que permitirá um controle melhor das anotações e facilitará consultas e análises sobre elas.

O sistema de anotações deve oferecer ferramentas para capturar as anotações internas do *data warehouse* e apresentá-las aos pesquisadores utilizando uma interface amigável com tabelas, gráficos e desenhos que representem os elementos do genoma e seus relacionamentos. Da mesma forma, devem existir formas do pesquisador acessar as anotações externas via *links* pela interface do sistema.

Como as fontes de dados externas estão sempre sendo atualizadas com novas anotações, o sistema de anotação deve importar periodicamente, tais anotações para o seu *data warehouse*. Da mesma forma, ele deve ser capaz de re-executar os programas e inserir as anotações automáticas geradas no *data warehouse*. Isto é útil especialmente no contexto de projetos genoma em andamento. Neste caso, enquanto novos *reads* estão sendo gerados em laboratório, os programas de montagem de fragmentos irão criar novos contigs (ou cDNAs completos, no caso de projetos genoma ESTs). Com isso, os programas que têm contigs ou cDNAs como entrada, também gerarão resultados diferentes. Assim, é importante que o sistema de anotação tenha controle de versões das anotações e seja capaz de transferir as anotações manuais adicionadas pelos pesquisadores entre as diferentes versões.

Um sistema de anotação também deve oferecer mecanismos de controle de execução de programas que estejam em *sites* externos ou locais. No caso dos programas locais, o sistema pode executá-los *offline* e armazenar seus resultados no *data warehouse*. Isto facilita o acesso aos resultados dos programas, mas obviamente exige que os dados de entrada já sejam conhecidos *a priori*, além de espaço em disco suficiente para armazenar os resultados. Esta estratégia é útil quando a comunidade de pesquisadores planeja acessar, visualizar e analisar estes

resultados repetidas vezes. Por outro lado, o sistema pode permitir a execução *online* dos programas sem armazenamento dos resultados em *data warehouse*. Isto torna-se interessante quando o dado de entrada do programa não é conhecido *a priori* e quando não há necessidade de compartilhar os resultados do programa com outros pesquisadores.

Em se tratando de programas de análise, nota-se a importância da composição de programas usando conceitos de workflow, o que será tratado na próxima seção.

Neste ambiente, onde muitos pesquisadores participam do projeto e têm necessidade de compartilhamento das informações, o sistema de anotação deve estar em ambiente *Web* e ter controle de acesso aos seus dados, já que muitos deles podem não ser de domínio público.

Atualmente existem diversos sistemas de anotações, destacando-se BioNotes [Lemos, 2003d, e], Artemis [Rutherford et al., 2000], DAS [Pearson, 2004], EDITtoTrEMBL [Moller et al., 1999], GASP [Reese et al., 2000], GenDB [Meyer, 2003], GeneMine [Lee, 2001], GeneQuiz [Hoersch et al., 2000], Apollo [Lewis, et al., 2002], GBrowser [Generic Model Organism Project, 2004], Imagene [Medigue et al., 1999], MAGPIE [Gaasterland, 1996], Manatee [TIGR, 2004b], Pedant [Frishman et al., 2001], Visual Genome [Rational Genomics, 2004], *Community Annotation Project* [CGR, 2004], *Alternative Splicing Annotation Project* [Modrek, 2004], Genestream [IGH, 2004], *Cancer Annotation Project* [LBI, 2004], *Ensembl Genome Annotation Project* [Clamp et al., 2003] e *NCBI's Genome Annotation Project* [Agarwala, 2004]. Poucos destes sistemas atendem a maioria dos requisitos levantados anteriormente.

2.6 Integração

Atualmente um pesquisador tem acesso a um rico conjunto de fontes de dados, assim como uma variedade de programas de análise de dados.

Cada fonte de dados de Biologia Molecular existente contém informações de um pequeno domínio do conhecimento. O conhecimento completo está na união delas. Estas fontes estão distribuídas, são heterogêneas, possuem grandes volumes de dados em constante crescimento, possuem modelos de dados distintos, e armazenam seus dados como simples arquivos texto ou em sistemas de

gerenciamento de bancos de dados. Porém, geralmente, estas fontes não oferecem uma documentação detalhada do esquema do banco de dados.

Devido à heterogeneidade das informações, a coleta e integração em um conjunto de dados coerente é um grande problema. Um exemplo do desafio é o caso aparentemente trivial de como definir um gene. Para o GDB, um gene é um fragmento de DNA que pode ser transcrito e traduzido em uma proteína; para o Genbank e GSDB, no entanto, um gene é uma região do DNA contendo uma característica genética ou fenotípica, o que inclui regiões de DNA não codificantes como íntrons e promotores. Existe uma diferença semântica clara entre estes dois conceitos de gene.

Um dos grandes desafios da Bioinformática é a construção de ferramentas de integração das informações que residem nestas fontes de dados [Seibel, 2002]. Entre os projetos que tratam desta integração estão o *Sequence Retrieval System* [SRS, 2004], o CPL/Kleisli[Davidson et al., 2001], e o TAMBIS [Baker et al., 1998; Goble et al., 2001].

Outro desafio consiste no uso destes programas, a serem executados sobre os dados de uma base integrada, pois atualmente cada fonte de dados de Biologia Molecular admite o uso de um número reduzido de aplicações [Seibel, 2002]. O Bio-AXS [Seibel, 2001; Seibel, 2002; Seibel, 2003] é um exemplo de projeto que trata da integração tanto das fontes de dados quanto dos aplicativos de Biologia Molecular.

2.7 Ontologia

Para lidar com a heterogeneidade e evolução de inúmeros bancos de dados e programas de análise é necessário entender e definir os conceitos inerentes a este contexto. Isto pode ser feito de forma consistente e transparente através do uso de uma ontologia.

Uma ontologia é uma especificação de conceitos [Gruber, 1993; Guarino, 1998; Stoffel; 1997], ou seja, uma descrição concisa e não ambígua de quem são as entidades relevantes no domínio da aplicação e como elas se relacionam. As entidades podem ser objetos, processos, funções, predicados e outros tipos dependentes da aplicação [Schulze-Kremer, 1998].

Uma ontologia elimina as incertezas e más-interpretações da semântica dos bancos de dados, programas e seus relacionamentos e, consequentemente, facilita a criação de sistemas aplicados ao domínio da Bioinformática.

Neste cenário destacam-se dois tipos de ontologias:

• Ontologia de Biologia Molecular: identifica e associa os conceitos da Biologia Molecular, geralmente presentes em esquemas das fontes de dados existentes. Atualmente é possível destacar a Gene Ontology [GO, 2004], a TAO (Tambis Ontology) [Baker et al., 1999], a EcoCyc [Karp et al., 2002], a RiboWeb [Altman et al., 1999] e a MBO [Schulze-Kremer, 1998].

• Ontologia de processos de Bioinformática: define os conceitos dos programas de análise de dados da Biologia Molecular. Entre estes conceitos estão as entradas, saídas e utilização de cada programa. Atualmente é possível destacar a ontologia definida no myGrid [Stevens, 2003; Wroe, 2003].

2.8 Workflow

Muitas tarefas descritas pelos pesquisadores envolvem a composição de vários programas. Uma coleção de dados produzida por um programa pode, dada uma semântica apropriada, ser a coleção de entrada de outro programa.

Por exemplo, para predizer a estrutura tridimensional de uma dada proteína, o pesquisador pode executar o programa Modeller. Este programa tem como coleção de entrada, além da própria proteína, as estruturas que serão utilizadas como modelos para a determinação da estrutura. Neste caso, torna-se interessante fazer previamente uma comparação de tal proteína com as sequências armazenadas no PDB. Esta comparação indicará as sequências do PDB mais similares à proteína em questão cujas estruturas podem ser utilizadas como modelos, ou seja, coleção de entrada do programa Modeller [Cavalcanti et al., 2005].

No entanto, a composição destes programas não é uma tarefa simples de ser realizada pelos pesquisadores, tornando-se uma grande barreira para análises mais complexas e revelando a importância do uso de workflows em Bioinformática.

Um workflow diz respeito à automatização de procedimentos, onde documentos, informação ou tarefas são passadas entre os participantes de acordo com um conjunto pré-definido de regras, para se alcançar ou contribuir para um

objetivo global de um negócio. Apesar de um workflow poder ser manualmente organizado, na prática a maioria dos workflows são organizados dentro de um contexto de um sistema de informação para prover um apoio automatizado aos procedimentos [WfMC, 1995; WfMC, 1999; WfMC, 2004].

Neste trabalho, um workflow trata da automatização de uma composição de programas, que analisam dados experimentais e que ajudam um pesquisador a interpretar tais dados.

Existem diversos pacotes de programas disponíveis para os pesquisadores em biologia. Estes pacotes já possuem *scripts* que definem workflows fazendo chamadas a programas, utilizando parâmetros *default*. Por exemplo, o pacote Phred/Phrap possui um *script* que, dado os cromatogramas, chama todos os programas que gerarão os *reads* e os contigs, incluindo alguns que o usuário pode não tomar conhecimento. Por exemplo, o cross_match é um programa chamado por este *script* que compara cada *read* a uma biblioteca de vetores e, caso encontre um vetor (ou parte deste) no *read*, retira a sequência de vetor do *read*.

Além destes *scripts* prontos, geralmente os pesquisadores utilizam linguagens de *scripts* para implementar os seus workflows, tendo em vista a facilidade que estas linguagens oferecem para fazer chamadas de programas. Entretanto, os *scripts* são muito específicos e difíceis de serem reutilizados e de se fazer manutenção. Por exemplo, o acréscimo e a remoção de programas, o acesso aos dados, o desenvolvimento de analisadores sintáticos, e a configuração de parâmetros geralmente não são tarefas triviais.

Sendo assim, torna-se importante a colaboração de profissionais de informática para auxiliar os pesquisadores na definição e execução de seus workflows, fato que já vêm acontecendo atualmente. Como exemplo, o sistema de anotação BioNotes possui dois tipos básicos de workflow definidos: um para projetos genoma completo e outro para projetos genoma de ESTs. Atualmente os pesquisadores do Riogene [DBM, 2004] estão utilizando o workflow para genoma completo para analisar e anotar a bactéria *Gluconacetobacter diazotrophicus* e o workflow para genoma de ESTs para anotar o *Rhodnius prolixus*. Seguem os dois esquemas destes workflows.

O primeiro é o de genoma completo (Figura 1). O primeiro passo é a chamada dos programas Phred e Phrap para gerar os *reads* e contigs a partir dos cromatogramas da bactéria, obtidos pelos laboratórios do Riogene.

O segundo passo trata da predição de genes, que é feita através do Glimmer, que obtém ORFs (*open reading frames*, em inglês), que são possíveis genes. Como o Glimmer pode gerar resultados falsos positivos e falsos negativos, os pesquisadores analisam outras informações para identificar um gene, como por exemplo a presença de tRNAs (um outro tipo de gene, não detectado pelo Glimmer), terminadores e sítios de ligação ribossomal. Para tanto os programas como tRNAScan [Lowe, 1997], que encontra tRNAs, Transterm [Ermolaeva et al., 2000], que procura terminadores e RBSFinder, que busca sítios de ligação ribossomal, são executados.

O terceiro passo trata da descoberta da função do gene, que é feita através da comparação da sequência da ORF com outras sequências já existentes em fontes de dados públicas. Atualmente as ORFs são comparadas, utilizando o programa BLASTP, ao banco de dados NR [NCBI, 2004c] e, utilizando o programa HMMPFam [Eddy, 2004], à coleção de famílias protéicas e domínios armazenadas no banco de dados PFam.

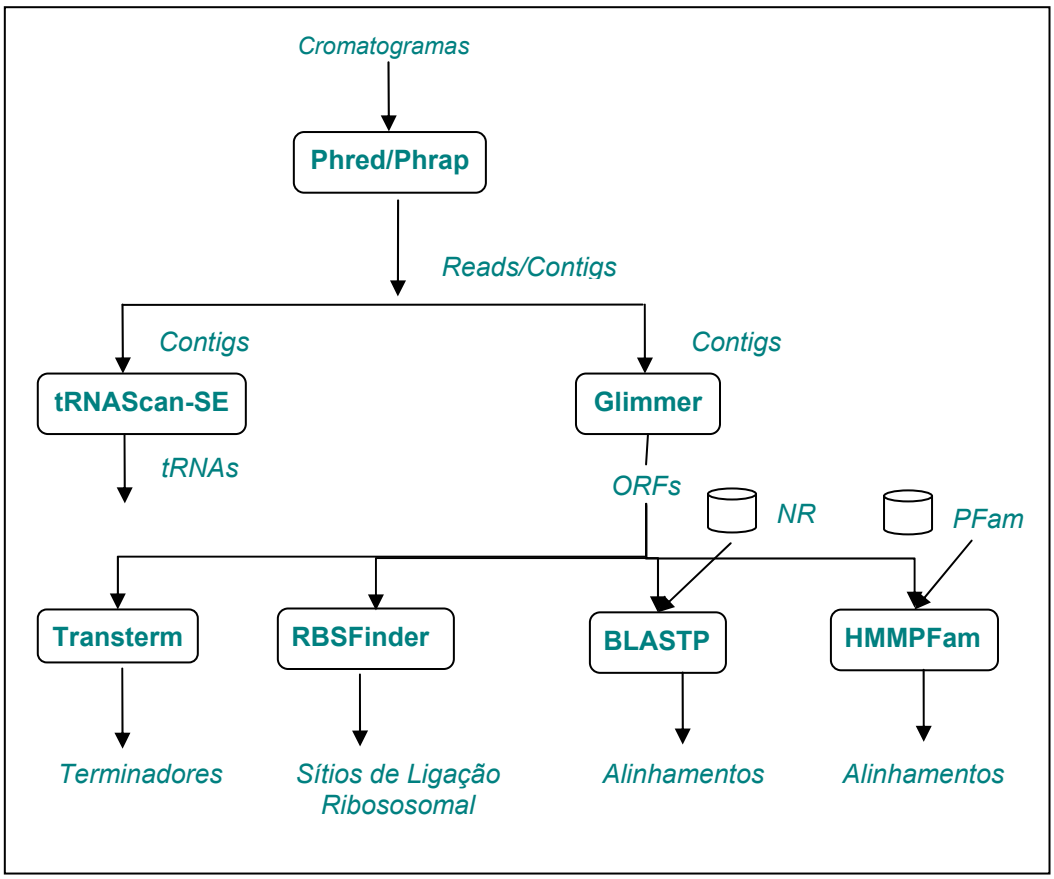

Figura 1. Esquema do workflow de projeto genoma completo.

O segundo esquema é um exemplo de workflow para projeto genoma de ESTs, usado atualmente para anotar o genoma do *Rhodnius prolixus* (Figura 2). Inicia-se com os cromatogramas, obtidos no laboratório, que são entradas do Phred. Os *reads* (sequências de nucleotídeos) gerados pelo Phred passam por dois filtros. O primeiro possui uma regra que elimina os *reads* que considera de baixa qualidade. Esta regra está baseada na qualidade de cada base do *read*, que também é uma saída do Phred. O segundo filtro retira vetores dos *reads*.

Com os *reads* de qualidade e sem vetor, existem mais três etapas. A primeira etapa inicia-se com o BLASTX, que faz a tradução da sequência de nucleotídeos do *read* nos seis *frames* possíveis, gerando seis sequências de aminoácidos, e realiza a comparação destas sequências com o banco de dados NR. Os seis possíveis *frames* são as fases de leitura em que o DNA é traduzido em proteína, através da "leitura" de trincas de nucleotídeos de forma sequencial. Isso faz com que seja possível a leitura de seis diferentes fases para a mesma molécula de DNA, sendo três fases para cada fita.

Diferentemente do projeto genoma completo, neste caso, os *reads* (em nucleotídeos) são partes da sequência cDNA, ou seja, partes do gene. Durante esta etapa, o pesquisador deseja descobrir a função do gene e, por isso, é importante a comparação com sequências armazenadas em outros bancos de dados. O BLASTX, além de mostrar sequências similares do NR, sugere qual é o *frame* correto do *read*. O programa GetCorrectFrame possui uma heurística para descobrir o frame correto a partir da saída do BLASTX e, de acordo com o frame descoberto, o programa Transeq faz a tradução do *read* para uma sequência de aminoácidos.

A sequência de aminoácidos é então entrada de outros programas que realizam sua comparação com outros bancos de dados.

Neste caso está sendo executado o RPS-BLASTN (reverse position specific BLASTN) [NCBI, 2004h] para fazer a comparação com o banco de dados CDD [Marchler-Bauer et al., 2003] e o SignalP [Bendtsen et al., 2004] para fazer comparação com um banco de dados de clivagem de peptídeo sinal, que são sequências localizadas na extremidade 5´ da proteína e que servem para direcionar a proteína em questão a um determinado local dentro ou fora da célula.

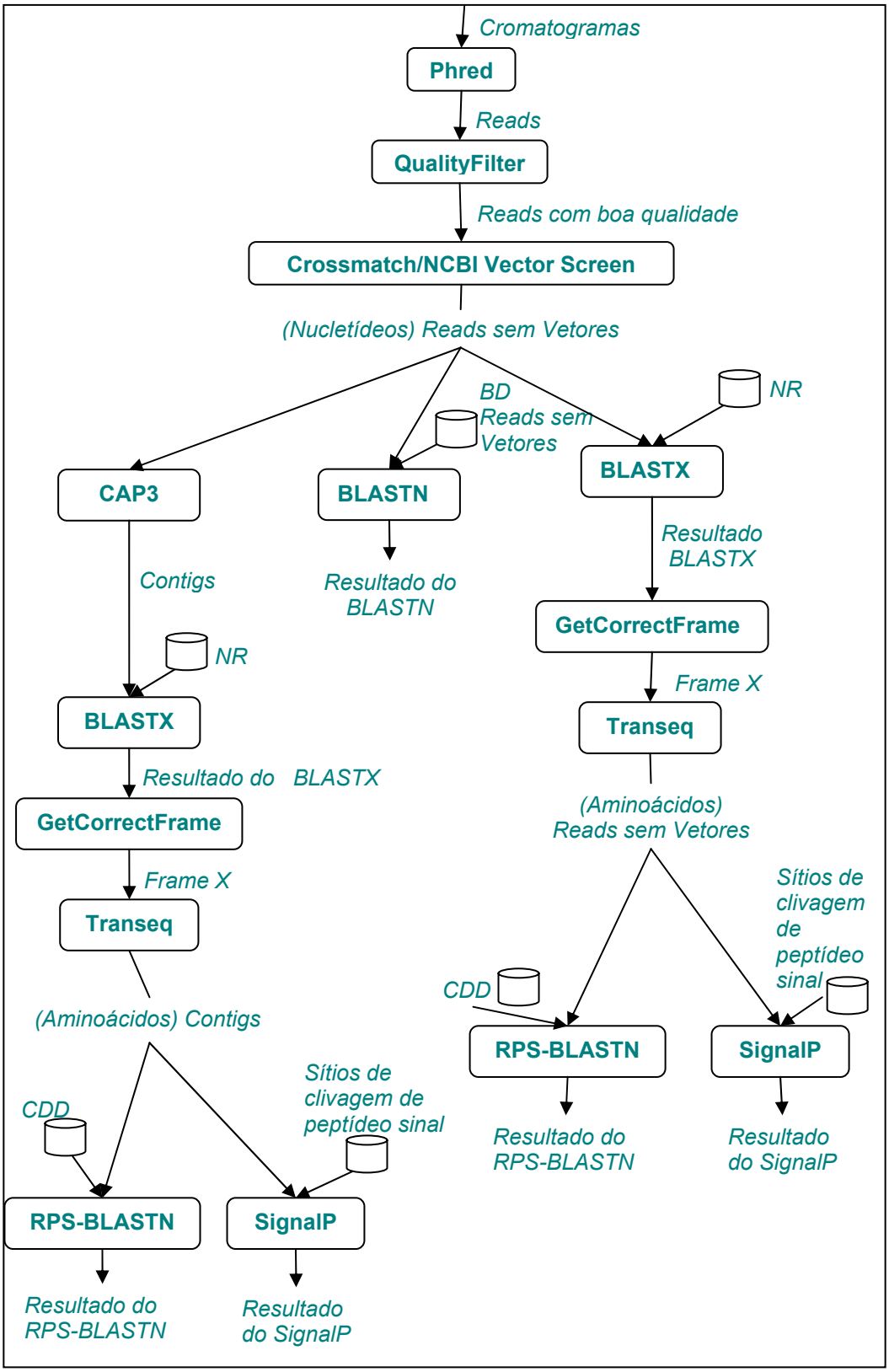

Figura 2. Esquema de workflow de projeto genoma de ESTs.

A segunda etapa trata da comparação dos novos *reads* (que acabaram de ser sequenciados) com todos os outros *reads* já sequenciados. Esta comparação é feita com o BLASTN e o objetivo é descobrir se o *read* novo já foi sequenciado antes. Esta etapa é importante para mostrar quando o processo de sequenciamento de um projeto genoma de ESTs não está mais produzindo sequências novas.

A terceira etapa é a montagem de fragmentos feita com o programa CAP3. No projeto genoma completo, esta etapa tem como foco a montagem de todos os *reads* formando contigs, sendo que se espera que, em algum momento do sequenciamento, todos os contigs sejam unidos formando o genoma completo ou, pelo menos, um cromossomo completo. No projeto genoma de ESTs, esta etapa trata da montagem de todos os *reads* que compõem um cDNA (também chamado de EST neste processo) ou parte de um cDNA. Ao traduzir o cDNA, obtém-se a proteína.

Depois da montagem, faz-se com o cDNA as mesmas comparações feitas com os *reads*, ou seja compara-se ao NR, CDD e SignalP.

2.9 Sistema de Gerência de Workflows de Bioinformática

Como a pesquisa em Bioinformática envolve vários programas de análise e bancos de dados, que estão em constante crescimento e atualização, é imensa a possibilidade de análises que podem ser feitas. Desta forma, para se ter produtividade, eficiência e qualidade nas análises, torna-se necessária a definição de uma linguagem de workflow apropriada com suporte de um sistema de gerência de workflows (SGW).

Um SGW oferece automatização dos procedimentos de um negócio através do gerenciamento de uma sequência de atividades de trabalho, e da invocação dos recursos humanos e tecnológicos apropriados, associados aos diversos passos da atividade.

A seguir está uma lista de requisitos que um SGW deve atender:

1. Processos, Dados e Recursos.

   O sistema deve incluir os processos, dados e recursos normalmente usados e oferecer mecanismos de extensibilidade para acomodar novos processos, dados e recursos. Entre os processos estão os programas de análise de

Bioinformática, programas que filtram os resultados de outros programas e mecanismos de controle de execução do workflow.

2. Definição.

O sistema deve ajudar os usuários na definição e redefinição do workflow. A redefinição é importante quando os resultados finais não forem considerados úteis ou interessantes pelos pesquisadores. Para tanto, o sistema deve permitir a definição de propriedades dos processos, dados e recursos de tal forma a facilitar a escolha dos mais adequados para cada caso exposto pelo pesquisador.

3. Validação.

O sistema deve oferecer ferramentas para validar o workflow definido pelo pesquisador. Durante a validação, o sistema deve verificar se as entradas e saídas definidas pelos usuários para cada processo do workflow são coerentes, além de incluir, caso seja necessário, processos que façam conversão nos formatos dos dados e processos que verifiquem se os resultados gerados pelos programas são esperados ou não. Este requisito é importante porque os programas de análise de Bioinformática nem sempre funcionam corretamente e geram resultados em formatos diferentes do esperado, o que possivelmente, implicará no não funcionamento de outros programas.

4. Otimização e Execução

O sistema deve ser capaz de otimizar e executar o workflow definido pelo pesquisador, de acordo com a arquitetura que está sendo utilizada.

A execução do workflow pode ser monitorada pelo pesquisador e deve permitir a intervenção do pesquisador. A intervenção é necessária caso o pesquisador queira avaliar os resultados intermediários para decidir se continua ou não a execução, ou para fazer alguma modificação na definição das próximas atividades do workflow.

5. Agendamento

O sistema deve oferecer agendamento da execução do workflow. O pesquisador pode desejar executar o workflow uma única vez em um determinado dia ou de tempos em tempos, como diariamente ou semanalmente.

6. Metadados

O sistema deve armazenar metadados sobre os workflows. Metadado é comumente definido como "dado sobre dado", ou ainda, de forma mais completa, como uma informação sobre o dado que permite o acesso e gerenciamento deste dado de maneira eficiente e inteligente [Sumpter, 1994][Dublin Core, 2004].

O sistema deve ser capaz de gerar estes metadados automaticamente, na medida do possível, e oferecer mecanismos para o usuário consultá-los e atualizá-los, quando for o caso. Os metadados ajudarão os pesquisadores a definir novos workflows, usar resultados gerados em execuções anteriores de workflows como entrada de novas execuções de workflows e minerar os resultados produzidos por execuções de workflows para descobrir informação.

2.10 Sistema de Gerência de Dados em Bioinformática

Além da gerência dos workflows em Bioinformática, é ainda necessário ter o suporte de um sistema de gerência de dados (SGD).

Este sistema deve ser capaz de armazenar, acessar e manipular as biossequências e todos os dados relacionados a elas, como as anotações automáticas, geradas como resultados dos processos durante a execução dos workflows; as anotações manuais, definidas pelos pesquisadores; e as anotações importadas, que são obtidas dos bancos de dados de Biologia Molecular.

Além disso, o sistema deve ser capaz de ajudar o pesquisador a analisar estes dados, por exemplo, através de ferramentas que permitam a criação de consultas sobre estes dados e a visualização dos resultados de forma amigável.

Como muitos projetos de pesquisa envolvem dados confidenciais, o sistema deve ainda garantir, através de mecanismos de segurança (como senhas), que os dados não serão acessados por pessoas não autorizadas.

2.11 Sistemas de Gerência de Análises em Biossequências

De acordo com o que foi argumentado neste capítulo, durante a análise de biossequências, os pesquisadores precisam de um sistema de gerência de workflows de Bioinformática (SGWBio) e de um sistema de gerência de dados em Bioinformática (SGDBio). Estes dois sistemas podem ser vistos como partes de um único sistema, denominado aqui de sistema de gerência de análises em biossequências (SGABio).

Esta tese apresenta uma proposta de um *framework* [Mattson, 1996] para um SGABio, que poderá ser instanciado em diversos ambientes de trabalho dos pesquisadores e que está baseado em uma ontologia de processos de Bioinformática.

Embora a tese apresente os dois sistemas que fazem parte do SGABio, será dada maior ênfase ao SGWBio, que está baseado em uma ontologia de processos de Bioinformática.

2.12 Trabalhos Relacionados

Atualmente existem alguns sistemas que tratam de gerência de workflows e dados em Bioinformática, sendo possível destacar o myGrid [Stevens, 2003], o Proteus [Cannataro et al., 2004], o Biopipe [Hoon et al., 2003], o LabFlow [Stein, 1995] e o Imagene [Medigue et al., 1999]. Entretanto, a maioria deles não contempla todos os requisitos que foram levantados anteriormente, como será descrito a seguir.

2.12.1 myGrid e Proteus

O myGrid é um projeto de desenvolvimento de uma camada *open-source* em alto nível, que pode ser utilizada por diversos laboratórios para a execução de aplicativos e workflows em um ambiente de grid [Foster, 2001; IBM Corporation, 2004].

Uma ontologia, definida em DAML-OIL [Horrocks, 2002; Wroe, 2003; W3C, 2001a, b], oferece um vocabulário controlado de conceitos que descrevem os serviços e workflows, tipos de entrada permitidas, tipos de saídas produzidas e recursos utilizados por estes serviços. Esta ontologia é utilizada para classificar e

recuperar serviços e workflows de acordo com a semântica dos dados de entrada e saída e, de acordo com as tarefas que cada serviço e workflow fazem.

Assim como o MyGrid, existe ainda um outro sistema implementado em uma arquitetura de Grid, chamado Proteus [Cannataro et al., 2004], que também é baseado em uma ontologia escrita em DAML+OIL. Ambos ajudam os pesquisadores na definição do workflow. Entretanto, nenhum dos dois possuem etapas de validação em que processos são incluídos automaticamente para que a composição dos processos definida pelo pesquisador se torne coerente ou otimizada. É necessário que o pesquisador conheça detalhes do ambiente de grid e os processos existentes para que ele mesmo defina otimizações em seu workflow.

Existem ainda alguns outros projetos de grid, incluindo o Asia Pacific BioGrid Initiative [APBI, 2004], o North Carolina BioGrid [The North Carolina Genomics and Bioinformatics Consortium, 2004] e o Bio GRID [EUROGRID, 2004]. O foco principal destes projetos é o compartilhamento de recursos computacionais, movimentação, replicação e análise dos dados em larga escala. Não é o objetivo atender os requisitos levantados anteriormente para um sistema de gerência de workflows.

2.12.2 Biopipe

O Biopipe [Hoon et al., 2003] é um *framework* com propósito de ser flexível, permitindo que o pesquisador defina *pipelines* através de módulos reutilizáveis.

O primeiro passo da criação do *pipeline* é a definição das tarefas que devem ser executadas e dos dados que serão manipulados de acordo com conceitos científicos. Por exemplo, o pesquisador pode dizer que gostaria de fazer um alinhamento múltiplo de sequências, e não que gostaria de executar o programa CLUSTAL W.

No segundo passo, o pesquisador define a composição das tarefas de acordo com a ordem das tarefas definidas no primeiro passo, e dos dados que serão utilizados como entrada e saída dos programas.

No terceiro passo, o pesquisador indica os programas para cada tarefa e a configuração dos parâmetros destes programas. Feito isso, o sistema gera um

documento de especificação do workflow em XML [W3C, 2004c] para ser executado.

De acordo com [Hoon et al., 2003], muitos dos requisitos anteriores não são tratados por este sistema, como, por exemplo, a validação e otimização do workflow. Além disso não é dito se o sistema possui uma ferramenta para ajudar o pesquisador no terceiro passo, ou se ele deve conhecer, *a priori*, qual programa é mais adequado em cada caso. Desta forma, não é sabido se o sistema possui definições de propriedades dos programas, dados e recursos que disponibiliza.

2.12.3 LabFlow

O LabBase é um banco de dados de Biologia Molecular específico de um laboratório e o LabFlow é um sistema de gerência de workflows. Estes dois componentes foram implementados separadamente para permitir mudanças no esquema do banco de dados e na definição do workflow, sem que uma interferisse na outra [Stein, 1995].

Para cada passo do workflow do laboratório, existe um pequeno programa, escrito em Perl, responsável por tratar as transações com o banco de dados e a análise aos dados. Desta forma, o LabFlow foi projetado como uma biblioteca de rotinas em Perl que são carregadas por cada *script* de processamento de dados durante o tempo de execução.

As descrições dos protocolos do laboratório são armazenadas em arquivos (*flat files*). Cada arquivo possui um ou mais workflows, cada um envolvendo uma sequência de passos do laboratório que afetam algum dado armazenado no LabBase.

Para modificar um workflow, é necessário modificar o arquivo de descrição dos protocolos em um editor de texto e alterar os *scripts* escritos em Perl.

De acordo com [Stein, 1995], não é o foco deste projeto a definição do workflow pelo pesquisador e, consequentemente, não é necessário a validação do workflow. Além disso, não é dito se há algum tratamento de otimização do workflow.

2.12.4 Imagene

O Imagene [Medigue, 1999] é um sistema de anotação que possui um gerente de dados, chamado de *Data Manager*, e um gerente de tarefas, chamado de *Task Manager*. O gerente de dados permite que o pesquisador crie e edite objetos associados aos dados, agrupe os objetos em diretórios e consulte o banco de dados de objetos. O gerente de tarefas é responsável pela escolha e disparo de estratégias (ou workflows) e por apresentar as tarefas e sub-tarefas durante a execução.

O principal objetivo do Imagene é ajudar o usuário a construir seu próprio workflow. Sendo assim, foram implementadas um grande número de tarefas elementares e um pequeno conjunto de composições de tarefas, chamadas de estratégias pelo Imagene. Entre as tarefas elementares estão as de acesso aos dados e as de análise de sequências, como predição de genes e alinhamento de sequências.

Existe uma ferramenta do Imagene que ajuda o pesquisador a criar estratégias mais complexas combinando tarefas e estratégias mais simples. Além disso, é possível modificar os valores dos parâmetros dos programas configurados inicialmente com valores *default*. Outra característica interessante do Imagene é a possibilidade de re-execução da estratégia a partir de qualquer ponto, para que o pesquisador possa modificar tarefas ou valores de parâmetros durante a execução.

O Imagene também possui uma ferramenta de visualização dos resultados dos programas.

A referência [Medigue et al., 1999], que descreve o Imagene, não comenta se é feita alguma validação no workflow definido pelo pesquisador. Além disso não é tratada a otimização do workflow definido pelo pesquisador.

2.13 Comentários Finais

Este capítulo mostrou uma discussão preliminar sucinta dos contextos biológico e computacional, apresentando uma visão geral dos assuntos em Bioinformática relevantes a esta tese, como os programas de análise, os bancos de dados, os sistemas de anotação, a integração dos dados e aplicativos, as ontologias, e os workflows.

De acordo com o que foi argumentado, mostrou-se a necessidade de um sistema de gerência de análises em biossequências (SGABio) que atendesse aos diversos requisitos expostos neste capítulo.

O Capítulo 5 apresentará então uma proposta de um *framework* para um SGABio que atende a todos os requisitos levantados anteriormente. O *framework* poderá ser instanciado para diversos ambientes de trabalho dos pesquisadores e está baseado em uma ontologia de processos de Bioinformática.

Este capítulo mostrou ainda trabalhos relacionados a um SGABio e ressaltou que nenhum deles contempla todos os requisitos levantados.