2
Fundamentos e Trabalhos Relacionados

2.1.
Banco de Dados Biológico

"Um Banco de Dados Biológico constitui um conjunto de dados, geralmente associado a um software projetado para atualizar, consultar e recuperar componentes dos dados armazenados no sistema" [Bioinformatics Factsheet 2004].

Banco de Dados Biológicos (BDB) ou Banco de Dados de Biologia Molecular (BDBM) são, geralmente, tabelas que possuem grandes quantidades de registros, onde seu significado é dado pela composição de informações de outros elementos. Por exemplo, um registro associado a uma sequência de proteínas contém normalmente uma descrição do tipo de molécula, seu nome científico e citações na literatura que correspondem a esta sequência.

O principal objetivo de um BDBM é permitir integrar e consultar, de forma otimizada, dados de seqüências de DNA, padrões de expressão de genes, estrutura de proteínas, conseqüências clínicas, dentre outros elementos resultantes de pesquisas efetuadas em um Projeto Genoma. Projeto Genoma é o nome de um trabalho conjunto realizado por diversos países visando desvendar o código genético de um organismo (podendo ser animal, vegetal, de fungos, bactérias ou de um vírus) através do seu mapeamento. Seu marco inicial é considerado o Projeto Genoma Humano.

2.1.1.
Características dos Dados Biológicos

Os dados biológicos apresentam muitas características especiais que dificultam o gerenciamento da informação biológica. A bioinformática trata do gerenciamento de informação genética com ênfase especial na análise da sequência do DNA, porém esta ainda é uma área que precisa ser estendida para um escopo mais amplo para controlar todos os tipos de informação biológica, como modelagem, armazenamento, recuperação e gerenciamento.

As principais características dos Dados Biológicos são [Elmasri and Navathe 2005]:

- Por serem altamente complexos, comparados a outras aplicações, as definições dos dados biológicos devem ser capazes de representar uma subestrutura de dados complexa e deverá garantir que nenhuma informação seja perdida durante a modelagem dos dados;
- Os sistemas biológicos devem ser flexíveis ao lidar com tipos e valores de dados. A colocação de restrições deve ser limitada, uma vez que isso pode excluir valores inesperados. A exclusão desses valores resulta em perda de informação;
- Os esquemas nos bancos de dados biológicos mudam muito rápido. Para um maior fluxo de informações entre gerações ou versões de bancos de dados, a evolução do esquema e a migração de objetos de dados devem ser possíveis. Um banco de dados evolutivo fornece um mecanismo oportuno e ordenado para acompanhar as modificações em entidades de dados individuais nos bancos de dados biológicos ao longo do tempo;
- Mesmo que se utilize o mesmo sistema, as representações dos mesmos dados por diferentes biólogos provavelmente serão diferentes. Desta maneira, devem ser suportados mecanismos para “alinhar” diferentes esquemas biológicos ou diferentes versões de esquema. Devido à complexidade dos dados biológicos, existem diversas maneiras para modelar qualquer entidade fornecida com resultados que refletem o foco particular do cientista. Ainda que dois biólogos produzam modelos de dados diferentes, se lhes for solicitado que interprete a mesma entidade, esses modelos provavelmente terão inúmeros pontos em comum. Nessas circunstâncias, é necessário que os pesquisadores sejam capazes de executar consultas através desses pontos comuns;
- A maioria dos usuários de dados biológicos não necessita de acesso de escrita no banco de dados, apenas acesso para leitura. Os usuários geram uma variedade de padrões de acesso de leitura no banco de dados que são diferentes dos padrões dos bancos de dados tradicionais;
- A maioria dos biólogos provavelmente não possui conhecimento da estrutura interna do banco de dados, ou seja, eles sabem de quais dados necessitam, mas não possuem conhecimentos técnicos sobre como um sistema de banco de dados representa os dados. Neste caso, as interfaces do banco de dados biológicos devem exibir para os usuários informações de maneira que seja aplicável para o problema que eles estejam tentando tratar e reflita a estrutura dos dados de bases;

- Os sistemas biológicos precisam dar suporte a consultas complexas, pois a definição e a representação destas consultas são extremamente importantes para o biólogo. Sem conhecimento da estrutura de dados, os usuários comuns não podem construir por conta própria uma consulta complexa através dos dados. Sendo assim, os sistemas devem fornecer ferramentas para que se construam essas consultas;
- Os pesquisadores desejam consultar os dados mais atualizados, mas devem também ser capazes de reconstruir trabalhos anteriores e reavaliar informações anteriores e atuais. Desta maneira, os valores que estão para ser atualizados em um BDB não devem ser descartados.

2.1.2. Tipos de Banco de Dados Biológicos

Os Bancos de Dados Biológicos (BDBs) ou Bancos de Dados de Biologia Molecular (BDBMs) podem ser classificados de acordo com o seu conteúdo. Este tipo de classificação é interessante e desejável especialmente por biólogos. No entanto, qualquer classificação de BDBMs conforme o conteúdo pode ser questionável do ponto de vista biológico. Normalmente cada biólogo tem sua própria classificação.

Em [Kröger, 2001] os BDBMs são classificados por conteúdo em nove grupos principais. Um banco de dados pode pertencer a mais de um grupo. Cada um destes grupos está descrito abaixo.

- Bancos de Dados Bibliográficos - resumem a literatura cientifica de uma forma legível para a máquina;
- Bancos de Dados Taxonômicos - trata-se de bancos de dados de classificação de espécies. São extremamente dependentes da classificação feita por um especialista;
- Bancos de Dados de Sequências de Nucleotídeos - enfocam entidades biológicas como genes e ácidos nucléicos. Em geral, visam o armazenamento e divulgação de dados de sequências de nucleotídeos de uma comunidade de pesquisa. As sequências de DNA e RNA são normalmente apresentadas juntamente com outras informações como o organismo a qual a sequência pertence ou ainda com as funções fisiológicas relacionadas à sequência;
- Bancos de Dados Genômicos - disponibilizam dados genéticos de um organismo especial, variando muito no conteúdo. As informações armazenadas em bancos de dados genômicos incluem informações

sobre genótipos, nome de genes, propriedades de genes, mutações específicas, assim como mapas genômicos e informações referentes a raças;

- Bancos de Dados Proteômicos - em geral, podem ser vistos como uma mistura de banco de dados de sequências de nucleotídeos, sequências de proteínas e outros.
- Bancos de Dados de Vias Metabólicas - armazenam informações sobre o metabolismo de um organismo ou de vários organismos diferentes. As enzimas participantes de reações são frequentemente relacionadas com bancos de dados de sequências;
- Bancos de Dados de Sequências de Proteínas - proporcionam informações sobre proteínas. Bancos de dados universais que armazenam informações sobre proteínas de todos os organismos. Devem ser diferenciados de bancos de dados especializados que armazenam informações sobre famílias especificas ou grupo de proteínas ou sobre as proteínas em espécies específicas;
- Bancos de Dados de Estrutura Proteica - estes bancos mantêm dados relativos à estrutura de proteínas. A estrutura 3D completa de proteínas é representada pelo armazenamento de coordenadas no espaço 3D;
- Bancos de Dados Híbridos - trata-se de bancos de dados que armazenam diferentes conteúdos, pertencendo a mais de um dos grupos citados.

Com o crescente número de dados biológicos que vêm sendo gerados, vários bancos de dados têm surgido. Anualmente a revista *Nucleic Acids Research* [NAR 2012], por exemplo, publica uma lista atualizada com a classificação de todos os bancos de dados biológicos disponíveis.

2.1.3. Formas de Armazenamento e Acesso

2.1.3.1. Armazenamento

Muitos Bancos de Dados (BDs) foram desenvolvidos no início dos anos 80, na época em que a Internet não era muito utilizada. Os dados eram

disponibilizados em formatos predefinidos, os *flatfiles*[1], e quando houve um aumento na sua utilização, passou a ser necessário o uso de *scripts* para a procura e recuperação de dados neste tipo de arquivo. Para a troca de dados, formatos de *flatfiles* predefinidos eram usualmente utilizados.

Esses *flatfiles* são arquivos do tipo texto estruturados utilizando códigos de letras no início de cada linha ou parágrafo, como pode ser visto na Figura 1, um exemplo adaptado de NCBI:

```
>gi|532319|pir|TVFV2E|TVFV2E envelope protein
ELRLRYCAPAGFALLKCNDADYDGFKTNCSNVSVVHCTNLMNTTVTTGLLLNGSYSENRT
QIWQKHRTSNDSALILLNKHYNLTVTCKRPGNKTVLPVTIMAGLVFHSQKYNLRLRQAWC
HFPSNWKGAWKEVKEEIVNLPKERYRGTNDPKRIFFQRQWGDPETANLWFNCHGEFFYCK
MDWFLNYLNNLTVDADHNECKNTSGTKSGNKRAPGPCVQRTYVACHIRSVIIWLETISKK
TYAPPREGHLECTSTVTGMTVELNYIPKNRTNVTLSPQIESIWAAELDRYKLVEITPIGF
APTEVRRYTGGHERQKRVPFVXXXXXXXXXXXXXXXXXXXXXXVQSQHLLAGILQQQKNL
LAAVEAQQQMLKLTIWGVK
```

Figura 1. Exemplo de flatfile

Este exemplo de *flatfile*, utilizado pelo NCBI, inicia com o símbolo '>' na primeira linha, indicando que esta corresponde à descrição da sequência que vem logo abaixo. Além disso, nenhuma linha tem tamanho maior que 80 caracteres.

Há alguns anos, um considerável número de BDBs estava baseado em soluções proprietárias de *flatfiles*, mas atualmente este número vem diminuindo e muitos destes têm sido substituídos por Sistemas de Gerência de Banco de Dados (SGBDs). Porém, representações com *flatfiles* ainda não são totalmente obsoletas, pois ferramentas de analise de sequência geralmente trabalham com esse tipo de arquivo.

De acordo com [Seibel 2000], os BDs implementados via SGBD possibilitam que cada laboratório utilize um BD distinto assim como podem adotar modelos diferentes, como por exemplo, modelos de dados relacional,

[1] Segundo [Stein 2004], *flatfiles* são arquivos de dados que contém registros com relacionamentos não estruturados. É necessário um conhecimento adicional sobre esses arquivos, por exemplo, as propriedades de seu formato, para que seja possível uma interpretação correta. Modernos sistemas de gerenciamentos de Bancos de Dados utilizam uma abordagem mais estruturada para a manutenção de seus arquivos.

orientado a objeto ou relacional-objeto. Cada um deles apresentando interfaces de consulta próprias, bem definidas.

Além dos *flatfiles* e SGBDs, outras formas de armazenamento que estão sendo bastante utilizadas no meio científico compreendem os *Object-Oriented Database Management Systems* (OODBMS) e a linguagem *eXtensible Markup Language* (XML).

Para [Köhler 2004], em torno de 7% dos BDBs são implementados utilizando OODBMS. Nesse tipo de armazenamento, qualquer tipo de dados pode ser implementado utilizando uma linguagem de programação orientada a objetos para gerar os objetos e os métodos específicos que serão utilizados para acessar e manipular os dados.

Ainda de acordo com [Köhler 2004], o XML está se tornando um padrão para troca de informações em BDBs, pois auxilia a superar a maioria das heterogeneidades. Muitas fontes de dados em *flatfiles* estão sendo convertidas para padrões XML, pois estruturas que apresentam muitos campos não específicos, como os campos característicos da maioria das bases de dados de *flatfiles*, são convertidos em *tags* XML apropriadas.

2.1.3.2. Acesso

A maioria dos BDs públicos pode ser acessada através de páginas Web. Essas páginas suportam pesquisas dos usuários às bases de dados. A maioria dos DBMS oferece aos usuários interfaces como o *Java Database Connectivity* (JDBC) e o *Open Database Connectivity* (ODBC), com as quais é possível pesquisar em BDs através de linguagens de pesquisa padrão como o *Structured Query Language* (SQL) e *Objetct Query Language* (OQL). Segundo [Köhler 2004] esses métodos são bastante utilizados na integração de bases de dados dentro das instituições. Por questões de segurança esses métodps são raramente utilizados para integração de BDs públicos. Nestes casos, a forma mais comum de troca de dados acontece usando XML e *flatfiles* via *http* ou *ftp*.

2.2. Modelos de Dados Tradicionais Usados em Bioinformática

Sistemas de Gerência de Banco de Dados (SGBDs) estão se tornando mais utilizados no setor de BDBM. Atualmente, os modelos de dados “tradicionais” mais referenciados para desenvolvimento de bancos de dados de biologia molecular são: o modelo relacional, o modelo orientado a objetos e o modelo semi-estruturado (bancos de dados XML).

2.2.1. Modelo Relacional

Este modelo foi resultado de um estudo realizado por Codd, tendo por base a teoria dos conjuntos. O modelo foi apresentado num artigo publicado em 1970 [Codd 1970], mas só nos anos 80 é que ele foi implementado.

O modelo relacional representa o banco de dados como uma coleção de relações [Elmasri e Navathe 2005]. Cada relação pode ser vista como uma tabela, onde cada coluna corresponde a atributos da relação e as linhas correspondem às tuplas ou elementos da relação.

Um conceito importante em um banco de dados relacional é o conceito de atributo chave, que permite identificar e diferenciar uma tupla de outra. Através do uso de chaves é possível acelerar o acesso a elementos (usando índices) e estabelecer relacionamentos entre as múltiplas tabelas de um sistema de banco de dados relacional.

Essa visão de dados organizados em tabelas oferece um conceito simples e familiar para a estruturação dos dados, sendo um dos motivos do sucesso de sistemas relacionais. Certamente, outros motivos para esse sucesso incluem o forte embasamento matemático por trás dos conceitos utilizados em bancos de dados relacionais e a uniformização na linguagem de manipulação de sistemas de bancos de dados relacionais através da linguagem SQL.

2.2.1.1. Tipos de Dados

O padrão SQL define uma grande variedade de tipos de dados para representar e manipular as mais diversas informações, dependendo das características do dado a ser armazenado. Como o foco desta Tese é representar sequências biológicas e/ou dados básicos da biologia molecular (dogma central), a seguir são apresentados os tipos de dados que são mais utilizados para este tipo de representação. No modelo relacional temos as seguintes alternativas: BLOB e VARCHAR.

2.2.1.1.1. BLOB

BLOB (*Binary Large Objects*) são objetos de dados que armazenam qualquer tipo de informação. Por esta característica, ele é utilizado principalmente para armazenar informações multimídias, e.g. músicas e vídeos, no formato binário em colunas de tabelas de banco de dados.

Outra vantagem do tipo BLOB é que não possuem limite de tamanho de armazenamento. Os valores de tamanho dos campos, que em outros tipos de dados são declarados no tipo, para Blob não existe esta necessidade de declaração, pois o tamanho dos campos é determinado pelo tamanho da página de dados informados no momento da criação do BD. Um vídeo WMF, de 17MB, por exemplo, é perfeitamente armazenável nesse tipo de campo, sem nenhuma restrição. O arquivo é armazenado diretamente no disco onde está a tabela de banco de dados, sem necessidade de fazer como algumas implementações de SGBDs que armazenam somente o nome e a extensão do arquivo em um campo, porém fisicamente o arquivo está armazenado em outra área do disco que não faz parte do banco de dados.

Outro ponto importante a se considerar é que estruturas do tipo BLOB não possuem mecanismos de manipulação de dados. A manipulação e aplicação de algum tipo de operador deve ser realizada na camada acima do SGBD, com o uso de alguma linguagem de programação.

2.2.1.1.2. VARCHAR

Varchar é um tipo de dados para armazenar sequências de dados ASCII (caracter/string) podendo ser de tamanho fixo, CHAR(n) ou CHARACTER(n), onde *n* é o número de caracteres, ou variável, VARCHAR(n) ou CHAR VARYING(n) ou CHARACTER VARYING(n), onde *n* representa o número máximo de caracteres [Elmasri e Navathe 2005].

VARCHAR e CHAR podem armazenar cadeias de até *n* caracteres de comprimento, onde *n* varia de acordo com a implementação do SGBD, como por exemplo, no Oracle 11g o limite máximo é de 4000 bytes ou caracteres para VARCHAR e 2000 bytes ou caracteres para CHAR. Caso seja atribuído um valor para uma coluna CHAR ou VARCHAR que exceda o tamanho máximo definido, o valor será truncado para o tamanho especificado. Por outro lado, se a sequência de caracteres a ser armazenada for menor que o tamanho declarado, os valores do tipo CHAR serão completados com espaços; e valores do tipo VARCHAR armazenarão simplesmente o tamanho da *string* fornecida.

Adicionalmente a estes dois tipos, mesmo não sendo um padrão SQL, muitos SGBDs implementam um tipo de dados para armazenar *strings* de

qualquer comprimento, e.g. o tipo *text* nos bancos PostgreSQL e MySQL. Como conjunto de funções inerentes a este tipo de dado temos[2]:

- **string || string** – concatenação de *string*;
- **bit_length(string)** – número de bits emu ma *string*;
- **char_length(string)** or **character_length(string)** – número de caracteres em uma *string*;
- **lower(string)** – converte uma *string* para caixa baixa;
- **overlay(string placing string from int [for int])** – substitui uma *substring*;
- **position(substring in string)** – localização de uma *substring* específica;
- **substring(string [from int] [for int])** – extrai uma substring;
- **upper(string)** – converte uma *string* para caixa alta.

Em campos VARCHAR podemos usar operadores como “=”, “>”, “BETWEEN”, “IN()”, “LIKE” (*case sensitive*), STARTING (*case sensitive*) e CONTAINING (*case insensitive*). Além disso, na maioria dos casos um índice pode ser usado para acelerar a busca dos dados. Já o tipo Blob não pode ser indexado, e os operadores e funções são restritos a alguns poucos oferecidos pelos SGBDs.

Outro ponto bastante relevante é a capacidade de utilizar campos do tipo VARCHAR para unir tabelas através do operador JOIN, fato este inexistente para campos do tipo Blob.

2.2.1.2. Vantagens e desvantagens

O armazenamento e gerenciamento de dados biológicos representam um desafio especial para bancos de dados relacionais, projetados para serem usados em cenários de dados tradicionais.

Dados biológicos são complexos. Um típico tipo de dado biológico tem uma estrutura aninhada de difícil representação no modelo relacional. Sistemas gerenciadores de bancos de dados relacionais frequentemente proporcionam um projeto fragmentado e não intuitivo [Kröger, 2001].

No contexto da biologia moldecular, o desafio inicial do modelo relacional é conseguir representar de forma intuitiva e verídica os conceitos e relacionamentos envolvidos. O grande problema é que a modelagem

[2] O conjunto completo de funções e operadores disponíveis pode ser visualizado na página do SGBD, neste caso o PostgreSQL [PostgreSQL 2012].

normalmente é realizada com informações que são descobertas e não como realmente elas são biologicamente.

O fato de decisões serem tomadas nos estágios iniciais, como a definição das entidades e dos atributos, caracteriza uma desvantagem do modelo relacional [Graves et al, 1995]. Uma vez que, em dados biológicos, não há como predizer qual fator se provará importante ou sujeito à modificação, e isto se torna um problema, pois alterações no esquema e implementação de um esquema são trabalhosas.

O modelo relacional é orientado a um eficiente armazenamento e gerenciamento de dados, mas não provê construtores para uma boa captura da semântica dos dados biológicos: a representação de um objeto conceitual complexo em um banco de dados relacional pode se estender por muitos registros em várias tabelas distintas [Markowitz et al, 1997]. O modelo relacional tem como ponto forte a capacidade de manipulação dos dados e busca de informação com o uso da linguagem SQL. Contudo mostra-se ineficaz para modelagem de objetos genômicos complexos [Shin 1995].

2.2.2. Modelo Orientado a Objeto

Assim como o modelo relacional, o modelo orientado a objeto está sendo empregado para o tratamento de dados biológicos. O INTERACT [Eilbeck et. al., 1999], por exemplo, um banco de dados sobre interações de proteína, utiliza o SGBD orientado a objetos *Poet*. Outros bancos de dados como o PSD/PIR [Wu et al., 2003], um banco de sequências de proteínas, também foram implementados usando um SGBD orientado a objetos.

2.2.2.1. Vantagens e Desvantagens do Modelo

No modelo orientado a objeto os dados são abstraídos e armazenados como objetos, possuindo estruturas com tipos pré-definidos. Sistemas orientados a objeto são melhores quando o esquema é complexo, o dado irregular e as consulta correlatas, sendo mais fácil pesquisar nas vizinhanças [Keet, 2003].

Uma vantagem de armazenar dados em um SGBD orientado a objeto é que ele é capaz de proporcionar uma melhor performance para dados complexos, e.g. dados biológicos. Além disso, sugestões podem ser incorporadas e novos métodos de bioinformática podem ser adicionados com um mínimo de código [McDermott and Samudrala 2002].

Modelos orientados a objeto enfatizam o comportamento de objetos e insistem que cada objeto tem sua própria identidade. Esta característica é de utilidade questionável em modelagem de grupos como colônias e desvia a atenção ao representar conceitos genômicos abstratos como experimentos, hibridizações ou localizações em um mapa [Graves et al, 1995].

A extensibilidade de sistemas de bancos de dados baseados em orientação a objeto também nos permite incorporar operações sobre os dados diretamente nas descrições de classe do objeto no banco de dados, deste modo escondendo os detalhes de implementação do usuário e permitindo ser usado diretamente com a linguagem de consulta do banco de dados.

A principal força do modelo orientado a objeto é seu poder de modelagem de dados altamente flexível, oferecendo uma maneira elegante de representação de objetos genômicos complexos. Por outro lado, não apresenta uma forma genérica de acesso aos objetos complexos [Shin 1995].

2.2.3. Modelo Semiestruturado (XML)

Pode-se definir dados semi-estruturados como dados que, apesar de não serem totalmente não estruturado, não são estritamente tipados. Em alguns casos, existe uma estrutura predefinida, em outros a estrutura está total ou parcialmente embutida no dado ou quase não apresentam informações descritivas. Ou seja, a informação que normalmente estaria associada a um esquema se encontra armazenada dentro dos próprios dados, os quais são por este motivo, muitas vezes, denominados auto descritivos. Isto significa que uma análise do dado deve ser feita para que a estrutura possa ser identificada e extraída.

Este tipo de dado tornou-se um importante tópico de pesquisas na área de banco de dados por diversas razões, dentre as quais destacam-se a proliferação de fontes de dados como a web e o desejo de desenvolver um formato extremamente flexível para troca de dados entre bases heterogêneas.

A modelagem de dados semiestruturados possui diferentes modelos de representação, entre os quais podemos citar o OEM (*Object Exchange Model*) e XML (*eXtensible Markup Language*).

2.2.3.1. Vantagens e Desvantagens do Modelo

Dados biológicos nem sempre são bem estruturados, muitas vezes se mostram incompletos, irregulares, redundantes ou contém erros. A maioria deles

são implicitamente estruturados. Portanto, dados da biologia molecular são bons candidatos para um modelo de dados semiestruturado [Kröger, 2001].

No modelo semiestruturado, o esquema é definido dinamicamente através dos dados (auto descritivo), apresentando uma descrição flexível de dados com relacionamentos complexos. A natureza auto descritiva de XML a torna uma forma promissora para definição de dados semi-estruturados [Shui et al, 2003].

Algumas características interessantes apresentadas por XML para aplicações em bioinformática, bem como algumas deficiências são apresentadas em [Achard et al, 2001] e descritas abaixo.

Quanto às características interessantes podemos citar:

- XML é altamente flexível, bastando modificar a DTD (*Document Type Definition*) ou XML Schema. Portanto, atualizar a definição de dados é uma tarefa simplificada.
- Como XML tem sua origem para representação, armazenamento e trafego na web, possui grande capacidade para vincular dados, podendo ser utilizado para interconectar bancos de dados *on-line*.
- XML proporciona uma área aberta para definir especificações padronizadas. Esta característica é um ponto importante, pois claramente há falta de padronização na bioinformática.

Por outro lado, XML apresenta algumas deficiências, tais como:

- O custo de um formato baseado em texto na analise de dados, armazenamento e transmissão precisam ser avaliados antes de adotar XML como uma solução geral. Contudo, um formato texto significa que o código fonte pode ser lido e editado com um editor de texto.
- A semântica de dados biológicos é bastante rica e requer um modelo de dados bastante expressivo. Embora possível, a modelagem de dados biológicos com XML é limitada, pois:
    - XML não tem mecanismos de herança e nem métodos em objetos.
    - O conceito de relacionamento pode ser imitado através de “referências fracas”, mas não existe como tal.
    - Existem poucos mecanismos para representar restrições de dados, e.g. restrições de unicidade, cardinalidade e não nulo;
    - XML não tem suporte para valores numéricos, tabelas e matrizes.

2.3. Dados Genômicos

Os genes são as unidades hereditárias em todos os organismos viventes. Eles constituem componentes essenciais do genoma (o conjunto completo de informação genética) desses organismos, sendo responsáveis pelo desenvolvimento físico, pelo metabolismo e, até certo ponto, pelo comportamento desses organismos. A maioria dos genes codifica para proteínas, grandes moléculas feitas de longas cadeias de moléculas menores chamadas aminoácidos, respondendo pela maioria das reações bioquímicas desempenhadas pelas células. Apesar da maioria dos genes especificar a construção de proteínas, alguns produzem moléculas de RNA muito importantes; outros não codificam para nenhuma molécula mas são importantes de um ponto de vista regulatório ou estrutural. Em qualquer dos casos, as moléculas produzidas como resultado da atividade de um determinado gene são conhecidas como produtos gênicos.

A informação contida no gene é a origem para a transmissão e expressão da hereditariedade entre os indivíduos. O processo que traduz o código genético em proteína é conhecido como dogma central da biologia molecular, proposto por Francis Crick em 1958 [Crick 1958] e divulgado em um artigo da revista Nature em 1970 [Crick 1970]. A Figura 2 ilustra este processo.

A transmissão da informação genética se inicia com o processo de replicação de moléculas de DNA. Um gene corresponde a uma região particular de uma molécula de DNA que pode abranger desde algumas dezenas de pares de nucleotídeos ou até muitas centenas. Cada gene codifica a produção de uma molécula específica de RNA, em um processo chamado transcrição gênica. Já a tradução corresponde ao processo de produção de uma proteína a partir de uma molécula de RNA. Este trio crítico de macromoléculas – DNA → RNA → proteínas - está presente em todas as células [Lodish et. al 2007].

Desde os anos 90, esforços internacionais levaram a determinação do código genético completo de mais de 400 organismos (http://www.genomesonline.org/), como bactérias, leveduras, parasitas protozoários, plantas, invertebrados e vertebrados, incluindo o homem (*Homo sapiens*). Mais de 1500 investigações genômicas estão em andamento, representando organismos de interesse comercial, ambiental, industrial, ou importantes modelos de pesquisa. Com a continuidade desses trabalhos, novas seqüências genômicas estão tornando-se disponíveis em um ritmo cada vez

mais acelerado, em adição a dados fragmentados de milhares de organismos, incluindo vírus. Os dados resultantes possuem o potencial de revelar os princípios básicos da genética, bioquímica e aspectos evolutivos desses organismos, assim como possibilitar o desenvolvimento de novos marcadores prognósticos, melhores medicamentos e vacinas, procedimentos diagnósticos aperfeiçoados, entre outros.

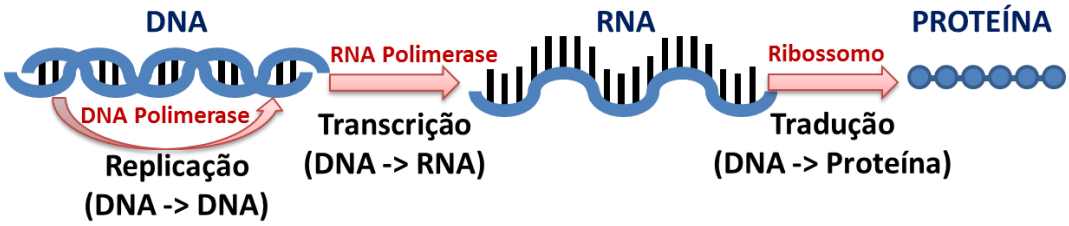

Figura 2. Dogma central da biologia molecular

Distintas partes do genoma codificam para as proteínas, as quais por sua vez dirigem as atividades funcionais e estruturais das células. Análises computacionais podem predizer quais regiões do genoma codificam para as proteínas. Entretanto, a predição das funções celulares destas proteínas (estruturais, enzimas, transportadores, sinalizadores, etc.) é em sua maioria hipotética. A maior parte dessas possíveis funções foi atribuída por análises *in silico* (em computadores), usando as técnicas de comparação de seqüências com bancos de dados de proteínas. Entretanto, até o momento, somente uma pequena fração das proteínas preditas teve suas funções confirmadas por experimentos laboratoriais.

A pesquisa científica e o desenvolvimento (bio)tecnológico baseados na genômica estão fazendo um progresso crescente no desenvolvimento de novos métodos diagnósticos, assim como no desenvolvimento de novas drogas e vacinas. A genômica comparativa e o conhecimento das vias bioquímicas e processos celulares são de enorme importância nessa área. Por outro lado, análises funcionais e estudos sobre as interações entre as proteínas são de extrema importância para a compreensão de como os microrganismos, das células em organismos multicelulares e patógenos interagem com seu ambiente e seus hospedeiros, abrindo caminho para o desenho de novas estratégias de controle para doenças infecciosas e parasitárias, assim como de doenças metabólicas, crônicas ou degenerativas.

2.3.1.
Análise Comparativa de Genomas

O desenvolvimento de métodos automáticos de seqüenciamento de DNA em larga escala, aliado ao desenvolvimento de tecnologias de computação de alto desempenho e de algoritmos mais eficientes, tem permitido à comunidade científica o uso de abordagens holísticas no estudo da estrutura, organização e evolução de genomas e na predição e classificação funcional de genes, através da análise de seqüências genômicas completas de centenas de organismos, particularmente procariotos.

Sequências genômicas completas constituem uma fonte de dados única, já que, em princípio, elas representam tudo o que é necessário para criar um organismo, juntamente com fatores epigenéticos e sua interação com estes. No entanto, não é imediatamente óbvio o que se pode fazer com toda esta informação.

A análise comparativa de genomas teve início com o seqüenciamento dos primeiros genomas na década de 1990. No entanto, suas ferramentas mais importantes têm origem nas técnicas clássicas de análise de seqüências: algoritmos de alinhamento global e local de pares de seqüências ou de múltiplas seqüências, métodos de análise filogenética e as implementações destes métodos e algoritmos, e.g. [Smith and Waterman 1981], [Lipman and Pearson 1988] e [Altschul et al. 1997]. De fato, ela se beneficia não somente de ferramentas desenvolvidas no passado, mas também da criação de novas ferramentas e do aperfeiçoamento das ferramentas já existentes, estimulados pela imensa, diversificada e complexa quantidade de dados produzida com os projetos de seqüenciamento em larga escala.

A etapa crucial deste tipo de análise é determinar se as seqüências comparadas são ou não homólogas, ou seja, se descendem ou não de uma seqüência ancestral comum, estabelecendo-se equivalência entre as partes comparadas. O resultado obtido permite, entre outras coisas, a predição de função, já que é presumido que seqüências homólogas tendem a ter funções similares [Bork and Koonin 1998] e também determinar quais os genes correspondentes entre os pares ou grupos de genomas analisados.

Esta tarefa nada trivial é feita comparando-se uma ou mais seqüências de entrada (*query sequences*), com outras inúmeras seqüências depositadas em um banco de dados (*subject sequences*), através do alinhamento consecutivo de cada seqüência de entrada com cada seqüência depositada no banco, com a

utilização de um algoritmo de alinhamento local [Smith and Waterman 1981], [Lipman and Pearson 1988] e [Altschul et al. 1997]. Para cada alinhamento, calcula-se o número de pontos obtidos (*score*), com base em uma Matriz de Substituição[3] - normalmente PAM (*Point Accepted Mutation*) [Dayhoff et al. 1978] ou BLOSUM (*BLOcks Substitution Matrix*) [Henikoff and Henikoff 1992] - e em valores arbitrados de penalidade para a abertura e extensão de espaços nas seqüências alinhadas (*gap opening/extension penalties*), e o número de alinhamentos esperados ao acaso com pontuação igual ou superior ao do alinhamento em questão (*E-value*), a partir da pontuação normalizada *(bitscore*) e do tamanho e composição do banco de dados. A homologia é inferida com base nos valores calculados dos diferentes parâmetros do alinhamento, alguns deles já mencionados: pontuação, pontuação normalizada, número de alinhamentos esperados ao acaso com pontuação igual ou superior ao do alinhamento em questão, percentual de identidade, percentual da extensão de cada seqüência no par alinhado que contribui para o alinhamento, diferença de tamanho entre as seqüências alinhadas etc. A existência de domínios (módulos que constituem unidades distintas do ponto de vista evolutivo, funcional e estrutural) em proteínas é um fator complicador nestas análises, que deve ser tratado com atenção.

A Figura 3, apresentada por Catanho [Catanho et al. 2007], representa genericamente os três níveis de abordagem da genômica comparativa de procariotos (e também de eucariotos) e algumas análises comumente realizadas. Uma vez que seqüências genômicas completas são obtidas através do seqüenciamento em larga escala dos genomas de diferentes espécies, análises comparativas envolvendo (i) a estrutura genômica, (ii) as regiões codificantes e (iii) as regiões não codificantes entre estes genomas podem ser realizadas, oferecendo múltiplas perspectivas acerca dos organismos estudados. Neste painel, segmentos genômicos sintênicos entre os genomas hipotéticos A, B e C são representados por barras horizontais de cores idênticas (Estrutura genômica). De maneira similar, regiões codificantes ortólogas (entre diferentes genomas) e parálogas (dentro de um mesmo genoma) são representadas por círculos de cores idênticas (Regiões codificantes). A presença de elementos regulatórios ou de pseudogenes, dentro de regiões não codificantes,

[3] São famílias de matrizes que contêm a probabilidade de que uma seqüência tenha sido transformada em outra durante o processo evolutivo.

conservados entre os genomas hipotéticos A, B e C são representadas por círculos pontilhados (Regiões não codificantes).

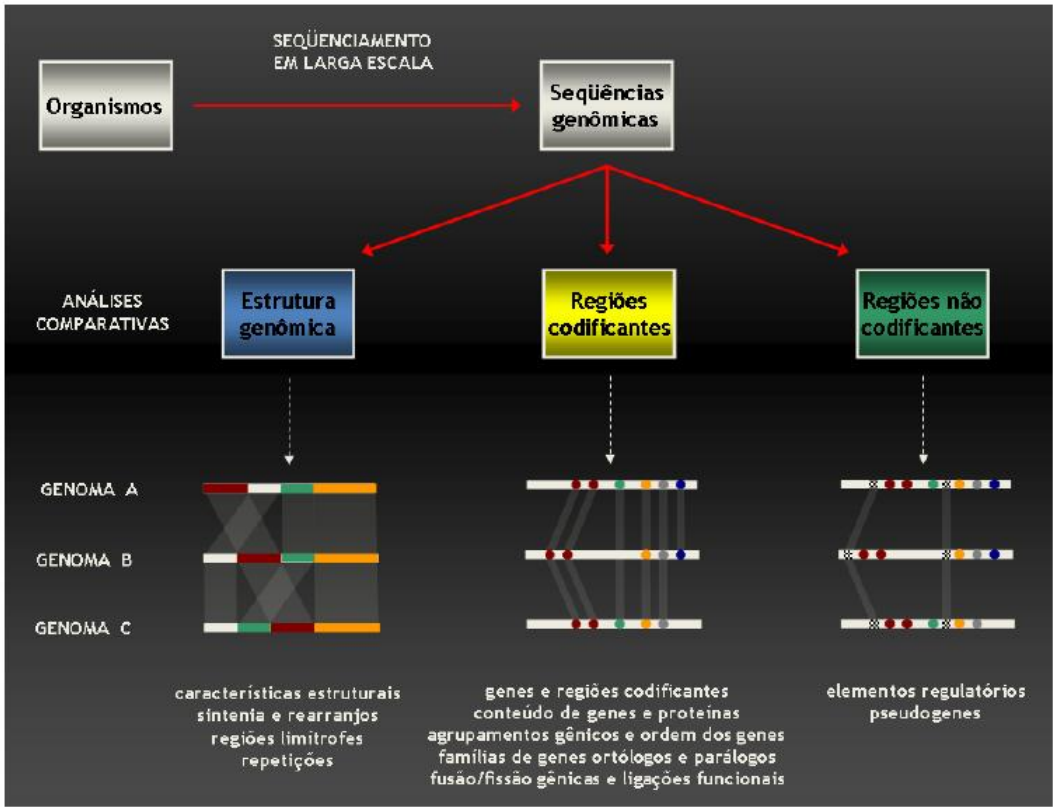

Figura 3. Análise comparativa de genomas

Fonte: [Catanho et al. 2007]

2.4. Trabalhos Relacionados

2.4.1. Projeto Comparação de Genomas

O projeto Comparação de Genomas é um projeto da equipe de Bioinformática do Laboratório de Genômica Funcional e Bioinformática do Instituto Oswaldo Cruz (IOC), FIOCRUZ, em parceria com o *World Community Grid* (www.worldcommunitygrid.org) e a equipe de Bioinformática do Laboratório de Bioinformática da PUC-Rio, com o objetivo de calcular o grau de similaridade entre sequências, comparando o conteúdo de proteínas de genomas completamente sequenciados de centenas de organismos, incluindo os seres humanos e várias outras espécies com grande importância na indústria, medicina, comércio, ou pesquisa, para posteriormente realizar a análise comparativa entre os genomas.

Para a realização do processo de comparação de proteínas foram utilizados os recursos de computação distribuída fornecidos pelo *World Community Grid*, possibilitando a distribuição de tarefas e otimização do processamento. Conforme já mencionado, a informação genômica, obtida no processo de comparação, pode ser usada para melhorar a qualidade e interpretação de dados biológicos, assim como a compreensão dos sistemas biológicos e das interações ambientais. Esta informação pode desempenhar um papel crítico no desenvolvimento de melhores medicamentos e vacinas, bem como métodos de diagnóstico.

As sequências utilizadas para as comparações no Projeto Comparação de Genomas foram sequências de aminoácidos obtidas de duas bases de dados distintas: o *Reference Sequence* (RefSeq) do NCBI [RefSeq 2012] (versão 21), um conjunto de sequências (que podem ser genômicas, mRNAs, RNAs, ou de proteínas) não redundantes e bem anotadas, e o SwissProt [UniProt 2012], uma base de dados de proteínas anotadas. Elas possuem organização e propostas diferentes com distintas formas de acesso.

Na primeira fase do projeto, utilizando o *World Community Grid*, foi realizada a comparação do tipo “todos contra todos” através do algoritmo de *Smith-Waterman* [Smith and Waterman 1981] em mais de 2,8 milhões de sequências de proteínas de aproximadamente 3.774 organismos, incluindo vírus, e dentre estes organismos, mais de 400 possuem a sequência completa do genoma decifrada. A maioria dessas sequências proteicas é originada a partir da análise computacional de genomas por parte de muitos grupos de pesquisa desde os anos sessenta. Elas são depositadas em bancos de dados públicos, juntamente com a anotação funcional, que na sua maioria são preditas computacionalmente. Para a análise comparativa de genomas, as sequências foram agrupadas em blocos contendo 2.000 sequências cada, e mais que 1 milhão de comparações do tipo bloco-a-bloco foram feitas. Posteriormente, numa nova rodada, foram realizadas mais de 4 milhões de comparações.

Para a segunda fase do projeto, o conjunto inicial de dados foi atualizado com dados genômicos mais recentes (RefSeq), sendo adicionadas 393.999 novas sequências proteicas. Além disso, um novo conjunto curado de dados de referência (SwissProt), representando mais 254.609 sequências, foi incluído nas comparações.

Finalmente, um conjunto de dados experimental representando mais de 3 milhões de sequências potencialmente codificantes foi adicionado na tentativa de identificação de novas sequências codificantes. Este conjunto de dados foi

derivado de *Open Reading Frames* (ORFs) com mais de 300 pb (pares de bases) para as quais uma predição clássica de seu potencial codificante não alcançou resultados positivos. Apenas as ORFs integralmente contidas em regiões descritas como sendo não-codificantes foram incluídas; qualquer tipo de sobreposição com uma sequência codificante anotada como tal resultou na exclusão da referida ORF do conjunto de dados analisados. Em termos de tamanho, os dados gerados a partir do Projeto Comparação de Genomas somam, compactados, aproximadamente 300GB; expandidos, estes dados ocupam quase 900GB de espaço em disco.

Ao todo, a tarefa de comparação de proteínas levou aproximadamente 5 meses de processamento no *World Community Grid*. Um exemplo de uma linha do resultado do programa SSEARCH pode ser visto na Figura 4. Somente a linha contendo os valores é armazenada. A linha superior contém os descritores dos valores: *query gi* (identificador da sequência *query*), *subject gi* (identificador da sequência *subject*); *SW score* (pontuação Smith-Waterman), *bit score*, *e-value*, *% identity* (percentual de identidade), *alignment length* (tamanho do alinhamento), *query start* (início da sequência *query*), *query end* (final da sequência *query*), *subject start* (início da sequência *subject*), *subject end* (final da sequência *subject*), *query gaps* (gaps na sequência *query*), *subject gaps* (gaps na sequência *subject*).

```
query gi, subject gi, SW score, bit score, e-value, % identity, alignment length, query start, query end, subject start, subject end, query gaps, subject gaps

67523787,67540134,2166,488.8,2.6e-138,0.336,1320,35,1275,67,1367,79,19
```

Figura 4. Resultado da execução do algoritmo de Smith-Waterman

2.4.2. Protein World DB

Conforme apresentado anteriormente, o projeto Comparação de Genomas gerou como resultado uma enorme quantidade de dados de similaridade entre proteínas. Com isso, gerou-se um novo problema a ser resolvido. Não havia uma estrutura/infraestrutura adequada para armazenar, gerenciar e dar suporte a pesquisas sobre estes dados. Neste momento teve início a colaboração do grupo de pesquisa em Bioinformática do Departamento de Informática da PUC-Rio. O objetivo desta parceria foi desenvolver um banco de dados para armazenamento dos dados resultantes do processo de comparação de proteínas, além de integrar dados de diferentes fontes públicas, e disponibilizar o acesso dos mesmos a toda comunidade científica.

Como solução e resultado da parceria entre Fiocruz e PUC-Rio, foi desenvolvido o banco de dados biológico *Protein World DB* [Otto et al. 2010], com o patrocínio da IBM e *World Community Grid*. Ele é o primeiro produto do Projeto Comparação de Genomas. Algumas de suas funcionalidades incluem a recuperação de identificadores, anotação, termos de ontologias e domínios de proteínas. Também é possível realizar pesquisas de similaridade utilizando a ferramenta BLAST.

Para o desenvolvimento do *Protein World DB*, várias questões foram levadas em consideração. Em primeiro lugar a questão de persistência dos dados, que eram da ordem de um terabyte. Sabe-se que não é suficiente apenas comprar mais dispositivos de armazenamento com maior capacidade para que se resolva a questão de acesso e busca eficientes. Os sistemas gerenciadores de bancos de dados (SGBD) ajudam no caso de bancos de dados ditos convencionais. Porém, essa questão ainda é um problema em aberto para a biologia.

2.4.3. Integração de Dados

Levando em consideração o grande volume, distribuição e heterogeneidade dos dados gerados pelos sistemas biológicos, existe na literatura uma variedade de trabalhos focados na integração destes dados. A principal estratégia é criar uma camada de abstração para acesso e manipulação dos dados. Para isso é utilizado algum tipo de mecanismo de integração, como por exemplo: arquitetura orientada a serviços, integração de *link*, *data warehousing*, integração de visão, arquiteturas orientadas a modelo, *workflows* e *mashups*. SBRML [Dada et al. 2010], BioDBnet [Mudunuri et al 2009] [bioDBnet 2012] e Bio2RDF [Belleau et al. 2008] são algumas propostas de mecanismos de integração para o domínio.

Systems Biology Results Markup Language (SBRML) [Dada et al. 2010], é uma linguagem baseada no padrão XML que associa um modelo com vários conjuntos de dados. Cada conjunto de dados é representado como uma série de valores associados com as variáveis do modelo, e os seus valores com os parâmetros correspondentes. SBRML oferece uma maneira flexível de indexar os resultados aos valores dos parâmetros do modelo, suportando tanto dados de planilhas quanto dados multidimensionais (cubo).

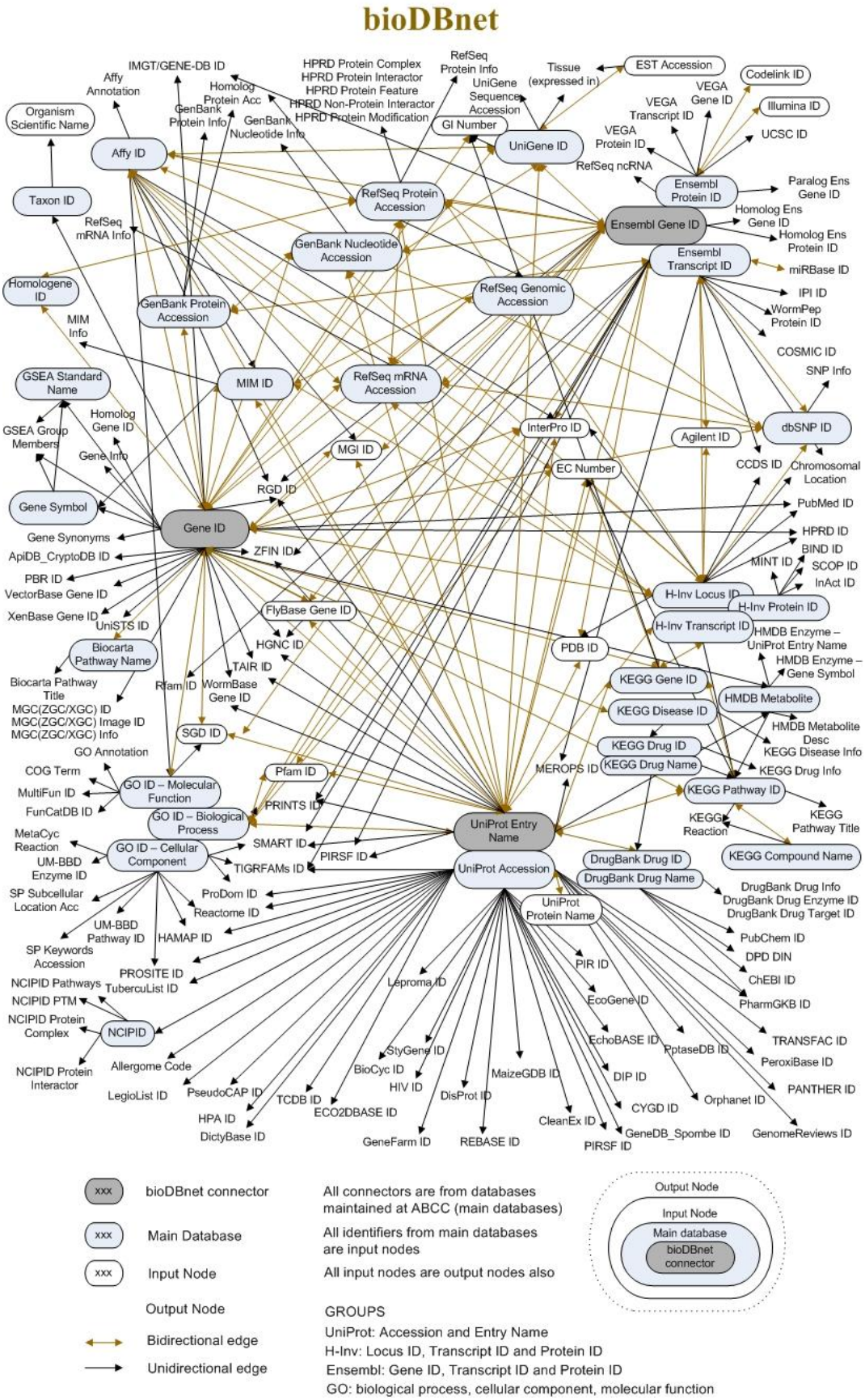

Figura 5. Conexões bioDBnet

Fonte: [bioDBnet 2012]

Biological DataBase network (bioDBnet) [Mudunuri et al 2009] [bioDBnet 2012] é uma aplicação web que integra uma variedade de dados biológicos, e.g. Gene, UniProt, Ensembl, GO, Affy, RefSeq etc. As bases de dados são geradas através de *download* de dados de vários recursos públicos, que são formatados e mantidos em uma estrutura relacional no *Advanced Biomedical Computing Center* (ABCC). bioDBnet integra mais de 20 bases de dados biológicas e faz referência cruzada de 189 distintos nodos, resultando em aproximadamente 663 conexões. A Figura 5, extraída do site oficial [bioDBnet 2012], ilustra estas conexões.

Já o Bio2RDF [Belleau et al. 2008] também é um sistema web que propõe um formato padrão para integração de várias bases de dados de bioinformática disponíveis em diferentes websites, como por exemplo, Kegg, PDB, MGI, HGNC e várias bases de dados do NCBI. Bio2RDF é um sistema *mashup*[4] que utiliza web semântica para interligar dados científicos, mais precisamente o padrão RDF (*Resource Description Framework*). A Figura 6 ilustra a arquitetura framework do sistema Bio2RDF.

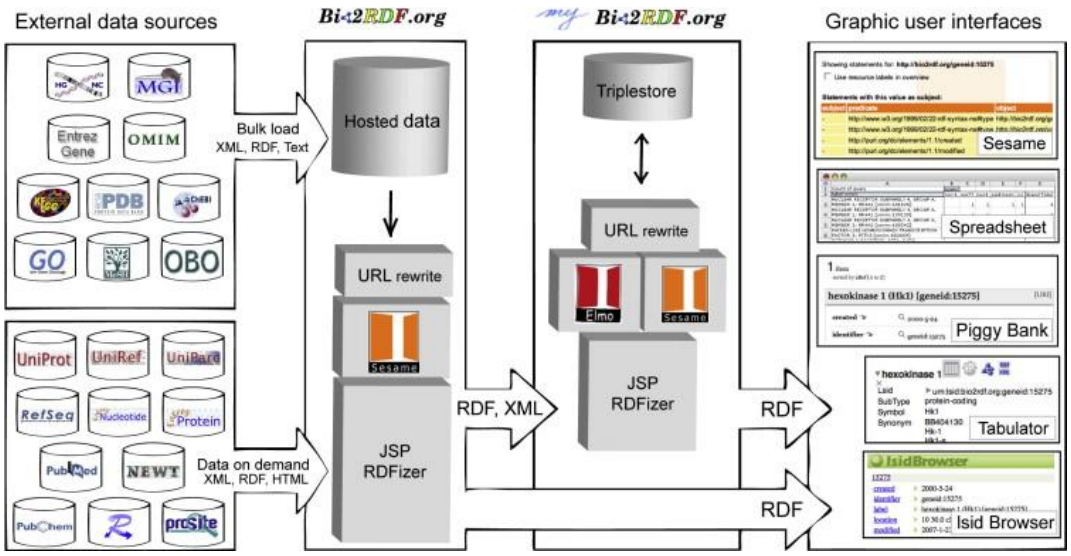

Figura 6. Bio2RDF: arquitetura framework do sistema

Fonte: [Belleau et al. 2008]

Normalmente no processo de integração os dados são coletados de fontes externas na forma de arquivos textos, armazenados em algum tipo de banco de dados, sobre um modelo biológico próprio, muitas vezes

[4] Um *mashup* é um site personalizado ou uma aplicação web que usa conteúdo de mais de uma fonte para criar um novo serviço completo.

desconhecido, e utilizando estruturas de persistência genéricas, e.g. string ou BLOB. O acesso e consulta a esses dados está condicionado a um conjunto de relatórios no formato texto e/ou ao dado bruto, ou seja, sem nenhum mecanismo de manipulação ou tratamento específico.

2.4.4. SGBDs e Extensões

Existem outros trabalhos que propõem a extensão de SGBDs relacionais incorporando ferramentas para prover acesso e manipulação de dados biológicos. É o caso do Oracle 10g[5] [Stephens et al 2005] e BLASTgres [BLASTgres 2010], que incorporam a ferramenta de alinhamento e comparação de bio-sequências BLAST (*Basic Local Alignment Search Tool*) [Altschul et al. 1990].

O módulo do SGBD Oracle versão 10g [Stephens et al 2005] que incorpora a implementação da ferramenta BLAST (NCBI BLAST 2.0) é denominado *Oracle Data Mining* (ODM) BLAST. Juntamente a esta ferramenta, é disponibilizado o módulo *Regular Expression Searches*, destinado a facilitar o trabalho de pesquisa na área da ciência da vida, fornecendo um grande número de funcionalidades para tarefas de pesquisa em texto e reconhecimento de padrões. Para poder usufruir das funcionalidades ODM BLAST, os dados de bio-sequências devem ser pré-carregados no Oracle Database 10g. Como benefício, o SGBD provê tabelas externas que permitem ao usuário realizar algumas consultas disponíveis em outros sistemas. Uma importante característica é que internamente as sequências biológicas são identificadas por um tipo de dados VARCHAR e a sequência propriamente dita por um dado do tipo CLOB (*Character Large OBject*). A Figura 7 ilustra a relação entre os componentes do SGBD Oracle 10g e a ferramenta BLAST.

Da mesma forma que o Oracle 10g, BLASTgres [BLASTgres 2010] [Hsiao et al. 2005] incorpora a ferramenta de alinhamento e comparação de bio-sequências BLAST ao SGBD PostgreSQL para suportar o gerenciamento de dados de sequências biológicas. Como diferencial, BLASTgres suporta um grande número de funções/operadores para manipulação dos dados biológicos e provê vários tipos de dados para representar informações de sequência no PostgreSQL. Os dois tipos mais importantes são: *range* e *loc*. *Range* é composto por um intervalo de inteiros que representa um segmento de sequência. *Loc*, por sua vez, é composto por um identificador de sequência e um range, e serve para

[5] Funcionalidade descontinuada no Oracle versão 11g.

representar um segmento de uma sequência específica. A Figura 8 ilustra a representação desses tipos.

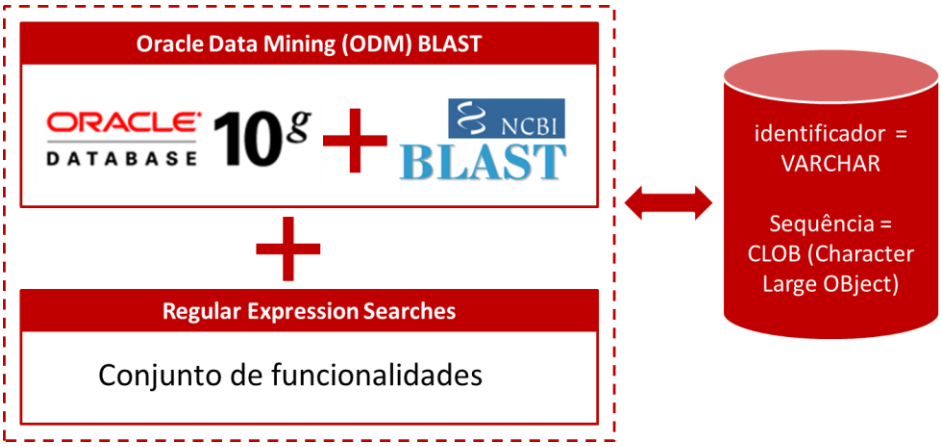

Figura 7. Oracle 10g x BLAST

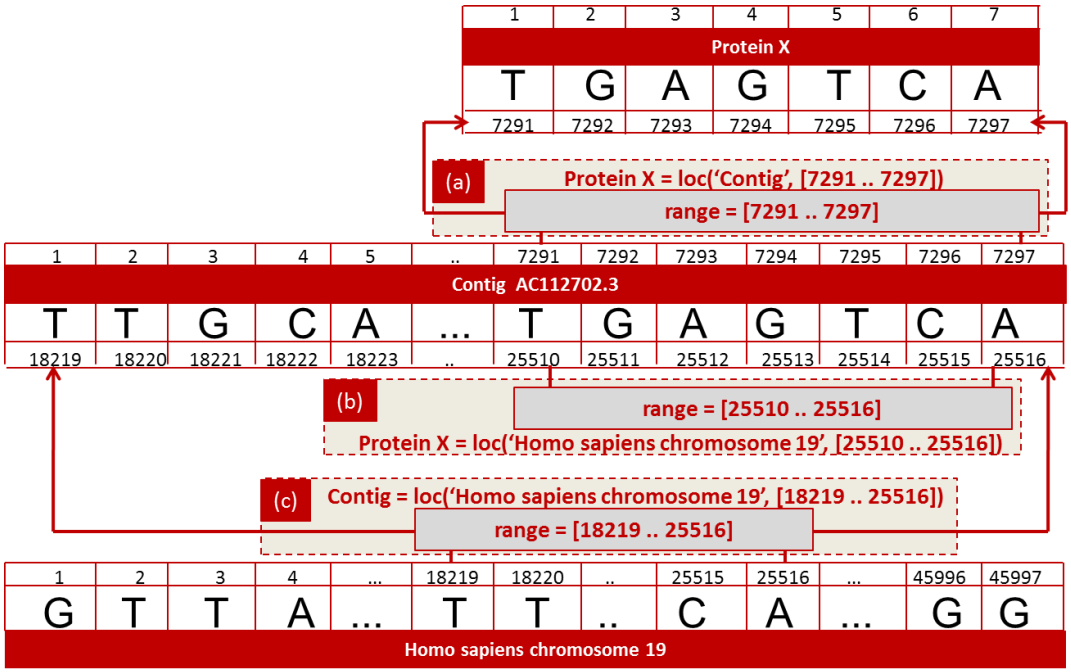

Figura 8. BLASTgres - representação de tipos loc e range

Loc está sempre associado a um range. Desta forma podemos definir a sequência contig AC112702.3 pela estrutura loc('Homo sapiens chromosome 19', [18219 .. 25516]), onde 'Homo sapiens chromosome 19' é o identificador da sequência e [18219 .. 25516] é o range (segmento) desta sequência que representa a contig AC112702.3 (Figura 8.c). Outro exemplo é a definição da proteina X, que pode ser feita de duas maneiras. Ela pode ser definida pelo range [7291 .. 7297] da sequência que representa o contig AC112702.3, conforme Figura 8.a - loc('AC112702.3', [7291 .. 7297]) - ou pelo range [25510 ..

25516] do cromossoma 19 da espécie Homo Sapiens, Figura 8.b - loc('Homo sapiens chromosome 19', [25510 .. 25516]).

Outra característica do BLASTgres é a possibilidade de parametrizar a consulta. O usuário tem disponível cerca de 50 parâmetros específicos. Para usá-los devemos definir em uma tabela, denominada *parameter table*, dois campos: um para o nome do parâmetro e outro para o valor. O resultado de uma consulta utilizando uma chamada BLAST é armazenado em uma tabela denominada *hit table*. Desta forma, os resultados BLAST podem ser visualizados e analisados, ordenando e/ou agrupando qualquer atributo, e podem ser integrados com outras informações biológicas do sistema. A Figura 9 ilustra a parametrização de consultas.

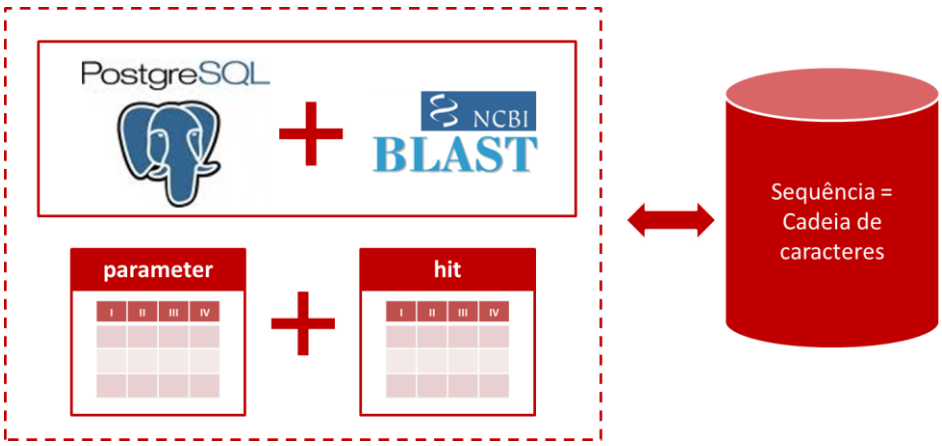

Figura 9. BLASTgres: parametrização de consultas

Contudo, da mesma forma que no Oracle 10g, as sequências biológicas não recebem um tratamento especial. Elas são armazenadas como simples cadeias de caracteres. Tanto o Oracle 10g quanto o BLASTgres possuem foco em abstração do uso da ferramenta BLAST por meio de consultas SQL e acesso dos resultados através de tabelas. Ambas abordagens restringem-se ao uso de uma ferramenta específica, objetivando ganhos de performance e melhorias na visualização e manipulação dos resultados. Nenhum deles apresenta uma forma de armazenamento dos dados em estruturas específicas e mais eficientes.

Outro trabalho muito interessante é o bdbms [Eltabakh et al. 2007] [Eltabakh et al. 2008b]. O bdbms é um protótipo de SGBD-Biológico extensível com foco no gerenciamento e proveniência de anotações de sequências biológicas. Para suportar o processamento e a pesquisa de informações sobre anotação e proveniência, o bdbms possui uma extensão do SQL, denominado *Annotation-SQL*, ou A-SQL para simplificar. Além disso, o bdbms possui um conjunto de funcionalidades e estruturas (tabelas) para gerenciamento das

anotações, como por exemplo, um mecanismo de autorização de *updates* hierárquico, denominado *content-based approval*.

Com relação ao armazenamento da sequência biológica, o bdbms não disponibiliza nenhuma estrutura de armazenamento ou mecanismos de manipulação específicos. No entanto, são disponibilizados métodos de acesso para diversos tipos de dados biológicos, e.g. sequências. Para possibilitar isso, foram inseridas novas estruturas de índices, tais como SP-GiST [Eltabakh et al. 2006] e o SBC-Tree [Eltabakh et al. 2008a]. SP-GiST é um framework extensível de indexação para suporte a dados multidimensionais, enquanto SBC-Tree (*String B-tree for Compressed sequences*) é uma estrutura para indexar e pesquisar sequências RLE (*Run-Length-Encoding*)-*compressed* sem descompactá-las.

Existem outros trabalhos que propõem estruturas de dados alternativas para indexar bases de dados sequenciais, tais como: q-grams [Navarro et al. 2000], suffix array [Manber and Myers 1993], LC-tries [Andersson and Nilsson 1995], String B-tree [Ferragina and Grossi 1999], prefix index [Jagadish et al. 2000] e suffix binary search trees [Irving and Love 2000].

2.5. Considerações Finais

Banco de Dados Biológicos (BDB) ou Banco de Dados de Biologia Molecular (BDBM) são altamente heterogêneos e os dados biológicos apresentam muitas características específicas que dificultam o gerenciamento de sua informação. Esta particularidade do domínio reflete na forma como os conceitos biológicos são representados. Cada laboratório ou pesquisador cria e organiza seus dados da maneira que acha mais conveniente, sendo este o grande fator para a geração de uma grande quantidade de bancos de dados heterogêneos.

Como podemos observar, já existem propostas para a gestão e manipulação de dados biológicos. Novos tipos abstratos de dados para armazenar informações biológicas, inclusão da ferramenta Blast em SBDS e uma forma de consultar os resultados por meio de consultas SQLs, índices sobre sequências genômicas, e integração de dados são todas alternativas que vão ao encontro de um “SGBD-Bio”. No entanto, nenhuma delas leva em consideração questões que envolvam a semântica de sequências biológicas. Não são apresentadas formas adequadas para manipular e obter informações relevantes.

Este é o grande problema envolvendo a gestão de dados em um BDB. Os dados são armazenados como texto “puro”, ou utilizando estruturas genéricas, como no caso de BLOB, ou inapropriadas, como por exemplo, o tipo string *core*. Faltam mecanismos apropriados, tanto para representar os dados biológicos quanto para manipular sequências genômicas.

Concluímos que o problema não está no modelo utilizado pra representar os dados, mas sim na falta de semântica das estruturas de dados existentes.

Vale salientar que o projeto *Protein World DB* representou um estudo de caso real. Foram evidenciados os problemas enfrentados para armazenar, acessar, visualizar e gerenciar grandes quantidades de dados, neste caso mais de 6,3 milhões de sequências envolvidas no processo de alinhamento e aproximadamente 1 TB de dados de similaridade.