3
Proposta de Tese

Conforme descrito no Capítulo anterior, existem na literatura inúmeros trabalhos na área de bioinformática envolvendo o uso de SGBDs, ferramentas amplamente utilizadas em outros domínios com recursos de gerência de dados já consolidados, para fins biológicos. Estes trabalhos propõem desde simples estruturas de integração até mecanismos mais eficientes de gerência de dados biológicos. Um aspecto interessante é que, na sua maioria, estes trabalhos são complementares. No entanto, até o momento, não foi encontrado na literatura, nem em produtos disponíveis, registros de propostas de trabalhos relacionados à modelagem, armazenamento e acesso específico de sequências biológicas e seus derivados. Uma sequência biológica não pode ser tratada simplesmente como um conjunto de caracteres. Ela possui interpretações e características bem específicas ao domínio e, no entanto, a sequência biológica e demais dados são modelados, armazenados e acessados como uma estrutura do tipo string ou BLOB *core*.

Levando em consideração as dificuldades existentes na gestão de dados biológicos e identificando a falta de estruturas de dados específicas para armazenamento e acesso de sequências, propõe-se como Tese de Doutorado um modelo conceitual biológico para representar informações do dogma central da biologia molecular, bem como um tipo abstrato de dado (ADT – do inglês *Abstract Data Types*) específico para a manipulação de sequências biológicas e seus derivados. Além disso, propõe-se a geração de mecanismos adequados para acesso e manipulação dos dados. O objetivo é propor um padrão para gestão de dados biológicos, algo como um SGBD-Bio. Desta forma, independente do projeto ou área de pesquisa, quando for usar um SGBD e necessitar representar e armazenar uma sequência biológica, todos terão os mesmos mecanismos para acesso e manipulação dos dados.

Para melhor entender, podemos fazer uma analogia com o tipo de dados *date/time*. Este tipo de dado poderia ser tranquilamente representado por uma string (09/07/2012) ou numérico (09072012). O grande problema é que, ao se fazer isso, um conjunto de informações é perdido. Os operadores existentes para manipulação de dados do tipo string e numérico não são aplicáveis a uma data,

como, por exemplo, a soma de duas datas. Por este motivo existe o tipo *date/time*. Ele possui uma estrutura particular de armazenamento e mecanismos e operadores específicos de consulta e manipulação. Um valor *date/time* é obrigatoriamente composto por números e caracteres especiais, e.g. “/”, “-” ou “:”, que representam a divisão entre dia, mês, ano, hora, minuto e segundo. Com relação à manipulação do tipo *date/time*, existe uma variedade de funções para conversão e acesso aos dados, como por exemplo, a função *day* que retorna o dia do mês (SELECT EXTRACT(DAY FROM TIMESTAMP '2012-07-09 10:08:40' = 09), e características específicas para operadores. Em tipos *date/time* o operador de adição, pode representar simplesmente a adição de alguns dias (date '2001-09-28' + integer '7' date = '2001-10-05') ou de apenas algumas horas (date '2001-09-28' + interval '1 hour' = timestamp '2001-09-28 01:00:00'), dependendo da necessidade e do uso da função.

Com relação ao estado da arte podemos elencar duas contribuições inovadoras, que se relacionam entre si:

- Tipo abstrato de dado (*string estendido*), que fornece um conjunto de funções apropriadas ao domínio biológico;
- Modelo conceitual biológico, que uma vez implementado, independente de tecnologia, permite obter um conjunto de informações biológicas e responder questões tais como quais são os genes únicos e genes homólogos (ortólogos e parálogos) de um indivíduo.

Vale lembrar que a alternativa de um sistema dedicado e com tipos e mecanismos específicos já é adotada em outras áreas de conhecimento, como por exemplo Sistemas de Informação Geográficas (SIG), e.g. [Dias et al. 2005] e [Neteler and Mitásová 2008], e Banco de Dados Temporais [Simonetto and Ruiz 2000].

3.1. Modelagem, Armazenamento e Acesso de Sequências Biológicas

A grande dificuldade em representar e manipular sequências biológicas está relacionada com sua origem. Em primeiro lugar, tudo que conhecemos hoje da biologia molecular são representações/abstrações de como as coisas realmente são. Em segundo lugar, uma sequência biológica, olhada isoladamente, não tem um significado próprio. A informação está “escondida” no conjunto de letras que compõe a sequência, sendo necessária uma análise/manipulação sobre a mesma para poder extrair tais informações.

Toda a informação genética de um organismo vivo está armazenada em sua sequência linear das quatro bases. Portanto, um alfabeto de quatro letras (A, T, C, G) deve codificar a estrutura primária (i.e., o número e a sequência dos 20 aminoácidos) de todas as proteínas. Só com este tipo de informação fica evidente que representar sequências biológicas, pura e simplesmente, como *blob* ou *string* não faz sentido. Todo conjunto de informação contido em uma sequencia biológica é perdido.

Além disso, analisando o comportamenro e alguns fenômenos (manipulação), representados pelo dogma central da biologia molecular, que uma sequência biológica pode sofrer, podemos definir algumas regras (R) e funções (F) que um tipo abstrato de dados deve considerar.

Na natureza existem dois tipos de ácidos nucleicos: DNA (ácido desoxirribonucleico) e RNA (ácido ribonucleico). Analogamente a um sistema de comunicação, essas informações são mantidas dentro da célula em forma de código, que no caso denomina-se código genético.

Em sua estrutura primária, os ácidos nucleicos (DNA e RNA) podem ser vistos como uma cadeia linear composta de unidades químicas simples chamadas nucleotídeos. Um nucleotídeo é um composto químico e possui três partes: um grupo fosfato, uma pentose (molécula de açúcar com cinco carbonos) e uma base orgânica. Nas moléculas de DNA a pentose é uma desoxirribose enquanto que nas moléculas de RNA a pentose é uma ribose. A base orgânica, também conhecida como base nitrogenada, é quem caracteriza cada um dos nucleotídeos, sendo comum o uso tanto do termo sequência de nucleotídeos quanto o termo sequência de bases. As bases são **A**denina, **G**uanina, **C**itosina, **T**imina e **U**racila, sendo as duas primeiras chamadas de purinas e as três últimas chamadas de pirimidinas. No DNA são encontradas as bases A, G, C e T. No RNA encontra-se a base U ao invés da base T.

R1 – toda sequência biológica estará armazenada sob a forma de nucleotídeo.

Moléculas de DNA compõem-se de duas fitas, que ligam-se entre si formando uma estrutura helicoidal, conhecida como dupla hélice. As duas fitas unem-se pela ligação regular das bases de seus nucleotídeos. A base **A** sempre liga-se a base **T** e a base **G** sempre liga-se a base **C**. Com isto, a sequência de

nucleotídeos numa das fitas determina completamente a molécula de DNA. É justamente esta propriedade que permite a autoduplicação do DNA.

F1 – Complemento: complement("sequence")

Dado uma sequência ela retorna o complemento da mesma.

Ex.: complement ('ACGGCTATTTAGAC') = TGCCGATAAATCTG

Cada fita do DNA tem duas extremidades, chamadas de 3' e 5', numa alusão aos átomos de carbono que ficam livres no açúcar que compõem cada nucleotídeo. As duas fitas são antiparalelas, ou seja, as fitas possuem orientação 5' 3' opostas uma em relação a outra. A convenção adotada mundialmente para representar moléculas de DNA é escrever apenas umas das fitas na direção 5' 3'.

R2 – toda sequência de nucleotídeo estará armazenada na direção 5' 3'.

F2 – Reverso: reverse("sequence")

Dado uma sequência ela retorna a sequência inversa.

Ex.: reverse ('ACGGCTATTTAGAC') = CAGATTTATCGGCA

Entre 1949 e 1953, Chargaff estudou detalhadamente a composição do DNA [Chargaff 1951] [Chargaff 1950]. Ele observou que, apesar da composição de bases variar de uma espécie para outra, a quantidade de adenina era igual à de timina (A = T) em todos os casos. Foi também notado que o número de bases de guanina e citosina era igual (G=C). Consequentemente, a quantidade total de purinas equivale à de pirimidinas (i.e. A+G = C+T). Por outro lado, a razão AT/GC varia consideravelmente entre as espécies.

F3 – Conteúdo GC: getGCcontent ("sequence")

Dado uma sequência, é retornado o conteúdo GC presente na sequência.

Ex.: getGCcontent('ACGGCTATTTAGACT') = 6

Na cadeia de nucleotídeos de DNA, um conjunto de 3 nucleotídeos corresponde a um aminoácido: são os tripletos. Através do processo de transcrição os tripletos de DNA são convertidos em códons de RNA. Estes códons são, à semelhança dos tripletos, conjuntos de 3 nucleotídeos da cadeia de RNA mensageiro.

F4 – Transcrição: transcript("sequence")

Dado uma sequência de DNA, é retornado o seu transcrito, ou seja, a sequência de RNA mensageiro.

Ex.: transcript('ACGGCTATTTAGACT') = ACGGCUAUUUAGACU

Este migra para o citoplasma da célula, onde se liga a um ribossomo e a uma molécula de RNA transportador. Através do processo de tradução e utilizando a informação genética do DNA do indivíduo com a molécula de RNA, o ribossomo produz então os aminoácidos para formarem as proteínas. A Tabela 1 apresenta os códigos dos aminoácidos e a Tabela 2 o sistema de tradução do código genético.

Tabela 1. Códigos dos aminoácidos

Nome	Abreviação	Letra
Glicina ou Glicocola	Gly	G
Alanina	Ala	A
Leucina	Leu	L
Valina	Val	V
Isoleucina	Ile	I
Prolina	Pro	P
Fenilalanina	Phe	F
Serina	Ser	S
Treonina	Thr	T
Cisteína	Cys	C
Tirosina	Tyr	Y
Asparagina	Asn	N
Glutamina	Gln	Q
Aspartato ou ácido aspártico	Asp	D
Glutamato ou ácido glutâmico	Glu	E
Arginina	Arg	R
Lisina	Lys	K
Histidina	His	H
Triptofano	Trp	W
Metionina	Met	M

Tabela 2. O sistema de tradução do código genético

		2ª base			
		U	**C**	**A**	**G**
1ª base	**U**	UUU (F) UUC (F) UUA (L) UUG (L)	UCU (S) UCC (S) UCA (S) UCG (S)	UAU (Y) UAC (Y) UAA (*Stop*) UAG (*Stop*)	UGU (C) UGC (C) UGA (*Stop*) UGG (W)
	C	CUU (L) CUC (L) CUA (L) CUG (L)	CCU (P) CCC (P) CCA (P) CCG (P)	CAU (H) CAC (H) CAA (Q) CAG (Q)	CGU (R) CGC (R) CGA (R) CGG (R)
	A	AUU (I) AUC (I) AUA (I) AUG (M), *Start*	ACU (T) ACC (T) ACA (T) ACG (T)	AAU (N) AAC (N) AAA (K) AAG (K)	AGU (S) AGC (S) AGA (R) AGG (R)
	G	GUU (V) GUC (V) GUA (V) GUG (V)	GCU (A) GCC (A) GCA (A) GCG (A)	GAU (D) GAC (D) GAA (E) GAG (E)	GGU (G) GGC (G) GGA (G) GGG (G)

F5 – Tradução: translation ("position", "sequence")

Dado uma sequência de nucleotídeos é retornada a sequência de aminoácidos. Para realizar este processo deve-se levar em conta a tabela de tradução do código genético.

Ex.: translation(2,'ACGGCTATTTAGACT') = RLFR

ORF (*Open Reading Frame*) é uma sequência de nucleotídeo em uma molécula de DNA que tem potencial para codificar um peptídeo ou uma proteína. Toda proteína é originada de uma ORF, mas nem toda ORF origina uma proteína.

Uma ORF é delimitada pelo códon de iniciação AUG, que codifica o aminoácido Metionina (Met), indicando que a sequência de aminoácidos da proteína começa a ser codificada ali, e pelos códons de finalização (UAA, UGA e UAG) que indicam à célula que a sequência de aminoácidos destinada àquela proteína acaba ali. Deste modo, todas as proteínas começam com o aminoácido Met. ORF que não possui o produto proteico identificado é chamada de URF (*unidentified reading frame*).

F6 – Procura ORF: searchORF ("position", "sequence", "tam")

Dado uma sequência de nucleotídeos é retornado um conjunto de ORFs (possíveis proteínas/subsequências) com um tamanho mínimo.

*Ex.: searchORF(1,'ACG**AUG**CUAUUUAGA**UAG**CUG', 10) =*

AUG**CUAUUUAGA**UAG

Este conjunto 2 regras e 6 funções já é suficiente para gerar uma enorme quantidade de informação. Além disso, facilita a tarefa de usuários que não possuam domínio, tanto na área de banco de dados quanto na área da biologia.

Com relação ao tipo abstrato de dados, temos duas alternativas: (I) a criação de uma nova estrutura que contemple todos estes requisitos e defina uma nova forma de armazenamento e manipulação dos dados, ou (II) estender um tipo de dados já existente, garantindo a existência destes requisitos e desenvolvendo apenas o necessário.

Se pensarmos em termos de implementação, tanto uma alternativa quanto a outra tem seus benefícios e desvantagens. A criação de um novo tipo tem a vantagem de se pensar e gerar uma estrutura e mecanismos apropriados/dedicados a este novo tipo de dado, podendo ter um desempenho superior a um tipo adaptado/estendido. Por outro lado, o esforço para geração deste novo tipo é superior, uma vez que é necessário criar todas as estruturas e mecanismos envolvidos para armazenamento e manipulação, além de ter que incorporá-lo ao sistema/solução existente. Já para o tipo estendido o cenário se inverte. Dependendo do tipo utilizado, a adapção pode ser simplificada, usando toda base já definida e desenvolvida. Por outro lado, em termos de desempenho pode ser que o resultado não seja tão satisfatório.

Para demonstrar que a dificuldade em gerenciar dados sequenciais biológicos não é um problema do modelo relacional, mas sim a falta de semantica nas estruturas de dados existentes, iremos propor um tipo "*bio-string*", que é uma adaptação/extensão do tipo "*text*" ou "*string*".

A escolha do tipo "*string*" deve-se a grande dificuldade em armazenar/representar sequências biológicas em estruturas do tipo *blob* por conta de sua inexpressividade. Como este tipo de estrutura é destinado ao armazenamento de qualquer dado, não existem mecanismos apropriados para acesso e manipulação do dado. Tudo é tratado como binário.

Já a estrutura de armazenamento do tipo *string* possui um padrão bem definido de armazenamento e mecanismos de acesso e manipulação de dados. Podemos utilizar a estrutura de armazenamento do tipo *string* para armazenar sequências biológicas, contudo devemos criar ou reescrever funções e/ou operadores específicos ao domínio da biologia molecular. Funções tais como *lower()*, *upper()* e *convert()* não fazem sentido para uma sequência biológica.

No próximo Capítulo será apresentado como este conjunto de regras e funções propostas foram implementadas e quais estruturas de armazenamento auxiliares foram necessárias ser desenvolvidas para permitir o acesso e manuseio dos dados.

3.2. Modelo de Dados

De acordo com [Elmasri and Navathe 2005] um modelo de dados é um conjunto de conceitos que podem ser usados para descrever a estrutura e as operações em um banco de dados. O modelo busca sistematizar o entendimento que é desenvolvido a respeito de objetos e fenômenos que serão representados em um sistema informatizado. No geral, os objetos e fenômenos reais são complexos demais para permitir uma representação completa, considerando os recursos à disposição dos sistemas gerenciadores de bancos de dados (SGBD) atuais. Desta forma, é necessário construir uma abstração dos objetos e fenômenos do mundo real, de modo a obter uma forma de representação conveniente, embora simplificada, que seja adequada às finalidades das aplicações do banco de dados.

Em um modelo de dados biológico, onde o objetivo é representar um conjunto de conceitos biológicos, esta tarefa de abstração é ainda maior. Em especial, a biologia molecular apresenta muitos conceitos que já são entendimentos abstratos de seu funcionamento e relacionamentos.

3.2.1. Modelo Conceitual

O modelo de dados conceitual visa demostrar, através de um diagrama em blocos, todas as relações entre as entidades envolvidas, e seus atributos. Para a representação do esquema foi utilizado um diagrama ER (*Entity-Relationship*) convencional, incluindo cardinalidades *min-max*.

No contexto da biologia molecular este não é um processo trivial. Na maioria das vezes não existe um consenso entre os biólogos com relação aos conceitos e como eles se relacionam. Não é difícil encontrar relacionamentos

entre “entidades” existentes pelo fato de desconhecer o “caminho” (relacionamento) que leva àquele conceito. Um exemplo é a descoberta de proteínas sem conhecer sua sequência genômica.

O domínio da biologia molecular é extenso e envolve uma grande variedade de conceitos. Deste modo, iremos restringir o domínio para representar os conceitos e relações envolvidos no dogma central da biologia molecular. Além disso, o modelo conceitual proposto possui informações de similaridade (*hits*) entre os aminoácidos (proteínas e ORFs) e de grupo taxonômico. Estes dados são fundamentais para obtenção/geração de informações referentes a genes únicos, genes homólogos e no processo de anotação.

Uma proteína é gerada a partir de um gene, que é uma região de uma sequência genômica. Um gene que codifica uma proteína é um transcrito e produz uma transcrição primária que, após algum processamento, gera um transcrito maduro contendo as sequências que codificam a proteína (CDS). O transcrito maduro é formado pela concatenação de subsequências contendo informação para proteínas (*exons*) e regiões não traduzidas (UTRs).

Uma ORF é uma série de nucleotídeos que apresenta um códon de início (AUG) e se estende até o primeiro códon terminal (UAA, UGA ou UAG). ORFs podem não ser codificadas em proteínas. Desta maneira, todas as sequências de codificação (CDS) são ORFs, mas nem toda ORF codifica uma proteína.

A entidade proteína representa a sequência de aminoácido de uma proteína com a sequência de nucleotídeos de um CDS, e o CDS com o gene e a sequência genômica que a contém, mantendo apenas uma referência externa a sua transcrição. Assim, o CDS é uma entidade cuja propriedade básica é manter o relacionamento entre as entidades proteína, gene e sequência genômica. Isto é feito através do posicionamento de uma dada região codificadora de gene (éxons) no sistema de coordenadas da sequência genômica que o contém. Cada éxon em um gene corresponde a uma subsequência CDS, definidos por uma posição inicial e final mapeados em um sistema de coordenadas de uma sequência genômica.

A sequência de nucleotídeo de um gene que codifica proteína é parte de uma sequência genômica possuindo sub-regiões de códons (éxons) e não-códons (íntrons e regiões não traduzidas). A leitura e transcrição de um gene gera o mRNA, que futuramente será processado e transcrito em uma sequência de aminoácido, ocorre em uma direção específica *in vivo* (5’ para 3’). *In silico*, quando considerado o sistema de coordenadas de uma sequência genômica

contendo o gene em questão, o sentido de leitura do gene será “+” se o gene for representado no *strand* com coordenadas crescente (*start* < *stop*), caso contrário “-” (ou complementar) quando o gene for representado no *strand* complementar da sequência genômica (*stop* < *start*).

A entidade gene também possui um identificador NCBI - Entrez Gene [Entrez Gene 2010], que é o *geneId*. A região de sua sequência genômica é definida por uma posição de início e de parada, um sentido de leitura, sua ordem em relação a outros genes na sequência genômica, um identificador de transcrição (a partir do RefSeq), e o conteúdo GC. Uma sequência de aminoácido ORF_T é análoga, e relaciona-se com a sequência de nucleotídeos genômica através de um *ORF_region* delimitada por uma posição de início e de parada no interior da sequência genômica, com o identificador de RefSeq da sequência genômica, o sentido de leitura, a sua posição em relação ao seu gene vizinho e a sequência em si.

Uma sequência de nucleotídeos genômica derivada a partir de um RefSeq contém os genes (contendo CDSs) que codificam para a sequência de aminoácido de proteínas. Estas sequências genômicas possuem um status, que refere-se ao estágio atual do projeto de sequenciamento. Os possíveis valores para essa propriedade são: “C*omplete*”, que tipicamente significa que cada cromossomo é representado por um único *scaffold* de uma sequência de qualidade muito elevada; “*Assembly*”, que tipicamente significa que *scaffolds* têm sido construídos não ao nível de cromossomo e/ou são de um projeto de baixa qualidade de sequenciamento, e em “*In Progress*”, que indica que tanto o projeto de sequenciamento está em fase de pré-montagem ou as sequências concluídas/montadas ainda não foram submetidas ao GenBank / EMBL / DDBJ.

Sequências genômicas RefSeq com o prefixo NC_ (moléculas de genoma completo incluindo genomas, cromossomas, organelas, e plasmídeos) incluem tanto aquelas obtidas pelo processamento automatizado quanto pela análise de peritos, e o sistema de coordenadas, o posicionamento e anotação de genes são mais estáveis. Prefixos NT_, NW_, NZ_ (*contig* ou *scaffold* e WGS inacabadas) indicam registros que não são revisados individualmente; as atualizações são liberadas em massa para um genoma. Anotações, montagem e posição do gene são provisórias. Essas sequências devem ser diferenciadas e cuidadosamente processadas. Relacionamentos *Protein*, *CDS*, *Gene*, *Genomic Sequence* podem ser incompletos ou mesmo ausentes.

A entidade *Genomic Sequence* possui um identificador RefSeq, definição e o comprimento da sequência, o tipo de molécula orgânica (DNA/RNA), *status*,

tipo de sequência (cromossomo, organela, plasmídeo), um identificador opcional do respectivo projeto genoma, conteúdo GC e um identificador do *taxon* original.

A taxonomia de organismos é um importante princípio de organização no estudo de sistemas biológicos. Herança, homologia por descendência comum, e a conservação de sequência e estrutura na determinação da função são todas ideias centrais da biologia que estão diretamente relacionados com a história evolutiva de qualquer grupo de organismo.

A classificação taxômica segue uma estrutura de hierarquia. Cada *táxon* é referido como um "nó". O "nó raiz" (taxid 1) está no topo da hierarquia. O caminho a partir do nó raiz para qualquer outro *táxon* em particular é chamado de "linhagem"; a coleção de todos os nós sob qualquer *táxon* particular é chamada de "sub árvore".

No modelo conceitual, o organismo do qual a sequência genômica foi derivada é o nó folha, definindo as espécies sequenciadas (ou um nível inferior). Ele contém o identificador taxID do NCBI (um identificador estável exclusivo para cada *táxon*), os nomes científicos e comuns, e sinônimos. Cada nó da árvore tem uma classificação (*rank*), um nó pai, e nós descendentes (ou não). A linhagem taxonômica pode ser obtida através de uma passagem de árvore de nós folha até a raiz.

Com relação à informação de similaridade, existem 3 possíveis combinações de *hits* envolvendo ORFs traduzidas e proteínas: (1) ORFs x ORFs; (2) proteínas x ORFs; e (3) proteínas x proteínas. A cardinalidade mínima para todas as relações é zero, caso a comparação não gere hits significante, e a cardinalidade máxima é *n*, pois podem existir vários hits significantes entre as comparações.

As sequências de aminoácidos traduzidas (ORF) são representadas por outra entidade - ORF_T – porque elas não possuem um identificador prévio. Informações sobre essas sequências incluem a referência para o organismo original, locatização e tamanho. Existem 3 tipos distintos de relacionamento entre hits, proteínas e ORFs traduzidas. Eles são definidos como:

1. hit_OO – resultado da comparação entre ORFs traduzidas;
2. hit_OP – resultado da comparação entre ORFs traduzidas com proteínas derivadas do SwissProt. As proteínas derivadas do RefSeq não foram utilizadas no processo de comparação com as ORFs traduzidas;
3. hit_PP – resultado da comparação de proteínas RefSeq com proteínas RefSeq e SwissProt.

Esses relacionamentos possuem atributos que especificam o resultado do processo de comparação, tomando como base as informações obtidas com a utilização do algoritmo Smith-Waterman, que são: *query gi, subject gi, SW score* (score bruto da comparação), *bit score* (score normalizado), *e-value* (significância do alinhamento), *% identity, alignment length* (tamanho do alinhamento), *query start, query end, subject start, subject end, query gaps, subject gaps*.

A Figura 10 apresenta uma visão geral do esquema conceitual estendido proposto. Vale salientar que os genes, os transcritos, ORFs e sequências genômicas são sequências de nucleotídeos, enquanto proteínas e ORFs traduzidas são sequências de aminoácidos.

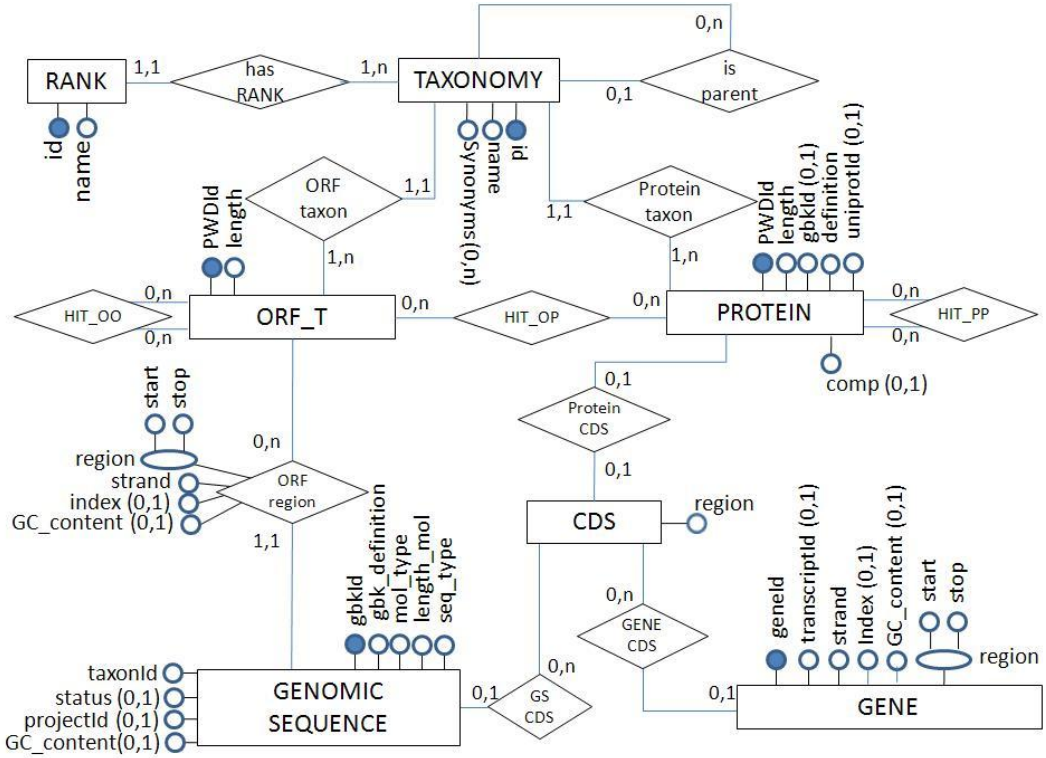

Figura 10. Diagrama conceitual

3.3. Considerações Finais

Neste Capítulo foi apresentada a proposta de modelo conceitual e estrutura para armazenamento de dados biológicos. Além disso, foi definido um conjunto de funções que manipulam e extraem informações biológicas de sequências, com base nas informações do dogma central da biologia molecular.

A ideia de propor uma modelagem conceitual biológica "genérica" ajuda a reforçar alguns conceitos biológicos, independente de pesquisas específicas ou

projetos. Com a implementação de um esquema lógico-relacional e a realização de algumas consultas (Capítulo seguinte), evidenciamos a eficácia do modelo.

O conjunto de regras e funções propostas mostrou a existência de informações semânticas intrinsicas no conjunto de letras de uma sequência biológica. Além disso, a ideia de implementar (Capítulo seguinte) este tipo abstrato de dados estendendo o tipo *string* serviu para mostrar que a dificuldade em gerenciar dados sequenciais biológicos não é um problema do modelo relacional, mas sim a falta de semantica nas estruturas de dados existentes.

O próximo Capítulo apresenta como tais funções foram implementadas e um estudo de caso envolvendo um cenário de teste para mostrar como o uso de SGBDs relacionais pode auxiliar e facilitar o processo de obtenção de informação.