Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Modelo de Dados para um Pipeline de Sequenciamento de Alto Desempenho Transcritômico

Ruben Cruz Huacarpuma

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Orientadora
Prof.ª Dr.ª Maristela Terto de Holanda

Brasília
2012

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Mestrado em Informática

Coordenador: Prof. Dr. Mauricio Ayala Rincón

Banca examinadora composta por:

Prof.ª Dr.ª Maristela Terto de Holanda (Orientadora) — CIC/UnB
Prof. Dr. Sérgio Lifschitz — Departamento de Informática/PUC-Rio
Prof.ª Dr.ª Célia Ghedini Ralha — CIC/UnB

CIP — Catalogação Internacional na Publicação

Huacarpuma, Ruben Cruz.

Modelo de Dados para um Pipeline de Sequenciamento de Alto Desempenho Transcritômico / Ruben Cruz Huacarpuma. Brasília : UnB, 2012.
99 p. : il. ; 29,5 cm.

Dissertação (Mestrado) — Universidade de Brasília, Brasília, 2012.

1. Modelo Conceitual, 2. Modelo de Dados, 3. Bioinformatica,
4. Banco de Dados, 5. Dados Biológicos

CDU 10/0055684

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Modelo de Dados para um Pipeline de Sequenciamento de Alto Desempenho Transcritômico

Ruben Cruz Huacarpuma

Dissertação apresentada como requisito parcial
para conclusão do Mestrado em Informática

Prof.ª Dr.ª Maristela Terto de Holanda (Orientadora)
CIC/UnB

Prof. Dr. Sérgio Lifschitz
Departamento de Informática/PUC-Rio

Prof.ª Dr.ª Célia Ghedini Ralha
CIC/UnB

Prof. Dr. Mauricio Ayala Rincón
Coordenador do Mestrado em Informática

Brasília, 01 de março de 2012

Dedicatória

Dedico este trabalho aos meus pais que, ainda longe, sempre acreditaram em mim. Exemplos de força e dedicação, bases da minha formação como pessoa, que cuidaram com atenção ensinando-me os valores da vida.

Aos meus irmãos e toda minha família que nunca se esqueceram de mim.

Aos mestres que souberam ensinar e guiar na direção correta, a todas as pessoas que acreditaram na minha capacidade. Em especial a minha orientadora, que ainda de culturas diferentes, soube me entender e acreditar em minhas habilidades, "MUCHAS GRACIAS" Dra. Maristela Holanda.

"Eu acredito demais na sorte. E tenho constatado que, quanto mais duro eu trabalho, mais sorte eu tenho." Thomas Jefferson

Agradecimentos

Antes de tudo preciso dizer que meus agradecimentos não são formais. Eu não me reconheceria neles se assim fora. Quero agradecer a todas as pessoas que se fizeram presentes, que se preocuparam, que foram solidárias, que torceram por mim. Mas bem sei que agradecer é sempre difícil. Posso cometer mais injustiças esquecendo pessoas que me ajudaram do que fazer justiça a todas que merecem.

De qualquer forma, todos os que realizam um trabalho de pesquisa sabem que não o fazem sozinhos, embora seja solitário o ato da leitura (em nossos tempos) e o do escrever. O resultado de nossos estudos foi possível apenas pela cooperação e pelo esforço de outros antes de nós. Como grandes pesquisadores da importância de Albert Einstein disse "Não descobri a teoria da relatividade apenas com o pensamento racional". Isto me leva a questionar quanto deste trabalho é meu e quanto é dos outros com quem convivi e com quem convivo, então chego à conclusão de que este trabalho não é só meu.

Queria agradecer de maneira especial a minha professora Maristela Terto de Holanda, minha orientadora do mestrado pelas aulas, pelas sugestões pelos conselhos e dicas de pesquisa, pelo material emprestado, pela paciência que teve comigo, pela participação e pela ajuda incondicional, juntamente com a Profesora Maria Emília M. T. Walter quem com seus conhecimentos e experiência souberam me encaminhar no mestrado. O professor Sérgio Lifschitz e a professora Célia Ghedini Ralha que são parte da minha banca de qualificação, agradeço pela sua presença, suas sugestões e contribuições para com meu trabalho.

Agradeço a todas as pessoas que confiaram em mim desde o primeiro momento que comecei o mestrado e me ajudaram nas minhas primeiras experiências neste novo país que me acolheu com braços aberto. Agradeço, particularmente, à Juliana Barbosa, minha primeira amiga e confidente no Brasil que fez todo o possível para eu me adaptar num lugar novo, de costumes diferentes dos meus, muito obrigado Juliana. Não poderia deixar de lado a minha família que, mesmo longe de mim, fez o possível para me ajudar e dar suporte nos momentos difíceis. Não poderia me esquecer de meus colegas de mestrado que me acompanharam nesta etapa da minha vida, muito obrigado Daniel Saad, Wosley Arruda, Tulio Conrado, Paulo Alvarez, Felipe Lessa, Halian Vilela, Taina Raiol, Beatriz Walter, Harley Olivera, e todo o pessoal da Bioinformática e do CIC com os quais passei bons momentos.

MUITO OBRIGADO A TODOS VOCÊS, NUNCA PODEREI PAGAR SEU APOIO SOMENTE COM MINHA GRATIDÃO ETERNA.

Resumo

O rápido avanço nas técnicas de sequenciamento de alto desempenho de fragmentos de DNA/RNA criou novos desafios computacionais na área de bioinformática. Um desses desafios é administrar o enorme volume de dados gerados pelos sequenciadores automáticos, particularmente o armazenamento e a análise desses dados processados em larga escala. A existência de diferentes formatos de representação, terminologia, estrutura de arquivos e semânticas, faz muito complexa a representação e administração desses dados. Neste contexto, um modelo de dados para representar, organizar e garantir o acesso aos dados biológicos é essencial para suportar o trabalho dos pesquisadores do campo da biologia, quando fazendo uso de *pipelines* de sequenciamento de alto desempenho.

Este trabalho propõe tanto um modelo de dados conceitual, como também seu respectivo esquema relacional, permitindo a representação e o gerenciamento de um *pipeline* de sequenciamento de alto desempenho para projetos transcritômicos no intuito de organizar e armazenar de maneira simples e eficiente os dados gerados em cada fase da análise do *pipeline*. Nesta dissertação, trabalhamos com *pipelines* de sequenciamento de alto desempenho com três fases: filtragem, mapeamento e análise. Para validar nosso modelo, apresentamos dois estudos de casos para identificar a expressão diferencial de genes usando dados de sequenciamento de alto desempenho transcritômico. Estes estudos de caso mostraram que introduzir o modelo de dados, e o esquema correspondente, tornou o *pipeline* mais eficiente, organizado, para dar suporte ao trabalho dos biólogos envolvidos em um projeto de transcritoma.

Palavras-chave: Modelo Conceitual, Modelo de Dados, Bioinformatica, Banco de Dados, Dados Biológicos

Abstract

The rapid advances in high-throughput sequencing techniques of DNA/RNA fragments created new computational challenges in bioinformatics. One of these challenges is to manage the enormous volume of data generated by automatic sequencers, specially storage and analysis of these data processed on large scale. The existence of representation format, terminology, file structure and semantics, becomes very complex representation and management of such data. In this context, a data model to represent, organize and provide access to biological data is essential to support the researchers works into biology field when using high-throughput sequencing.

This work proposes a conceptual model as well as its database schema to represent and manage a high-throughput transcriptome pipeline in order to organize and store in a simple and eficient way data generated in each pipeline phase. In this dissertation, we work with three phases high-throughput sequencing pipeline: filtering, mapping and analysis. In order to validate our model, we present two case studies both having the objective of identifying differentially expressed genes using high-throughput sequencing transcriptome data. These case studies showed that uses a data model, and its database schema, became the pipeline more eficeint, organized, and support the biologists works involved in a transcriptome project.

Keywords: Conceptual Model, Data Modeling, Bioinformatics, Database, Biological Data

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

BLOB	Binary Large Object Block
cDNA	Coded DNA
CDS	Doding Sequence
ChIP	Chromatin immunoprecipitation
COG	Clusters of Orthologous Groups
DNA	Deoxyribonucleic Acid
DOM	Dynamic Object Model
DoTS	Database of Transcript Sequence
EBI	European Bioinformatics Institute
edgeR	Empirical analysis of Digital Gene Expression data in R
EE	Espaço Economizado
EER	Enhanced Entity Relationship
EMBL	European Molecular Biology Laboratory
ER	Entity Relationship
GAI	Genome Analyzer I
GAII	Genome Analyzer II
GMOD	The Generic Model Organism Database
GUS	Genomics Unified Schema
KEGG	Kyoto Encyclopedia of Genes and Genomes
LNCaP	Lymph node Carcinoma of the Prostate
MCK	Molecular Computer Kit
MOO	Modelo Orientado a Objetos
mRNA	messenger RNA
NCBI	National Center for Biotechnology Information
ncRNA	non-coding RNA
PDB	Protein Data Bank
RAD	RNA Abundance Database
RefSeq	Reference Sequence
RNA	Ribonucleic Acid
SGBD	Sistema Gerenciador de Banco de Dados
SQL	Structured Query Language
SR	Source Shared
SRS	Short Read Sequences
TESS	Transcription Element Search System
TOAST	The Oversized - Attribute Storage Technique
tRNA	transfer RNA
UML	Unified Modeling Language

Capítulo 1

Introdução

Desde a descoberta da estrutura do DNA (*Deoxyribonucleic Acid*) em 1953, por Watson e Crick [13] os avanços na biologia molecular têm notáveis progressos criando-se uma nova área de pesquisa, a bioinformática. Um dos desafios que se destaca é o tratamento do grande volume de dados biológicos gerados pelos modernos sequenciadores de alto desempenho. Os gigabytes de sequências de DNA gerados por cada projeto de sequenciamento precisam ser armazenados. Enquanto o Projeto do Genoma Humano demorou 10 anos e custou aproximado de $3 bilhões de dólares [14] gerando aproximadamente 3.5 bilhões de pares de bases (pb), atualmente os novos projetos ficaram mais rápidos, baratos e geram maiores quantidades de dados (entre 2 a 4 bilhões de pb em poucos dias). Como exemplo desses projetos tem-se o "*Personal Genome Project*" de grande ajuda para área médica, com o objetivo de obter diagnósticos mais precisos de doenças e tratamentos médicos mais apropiados para um indivíduo particular [15]. Para este e outros projetos são usados os sequenciadores de alto desempenho, tais como Illumina [16].

A Bioinformática estuda os genomas que são compostos por cromossomos, cada um sendo uma cadeia longa de DNA de 4 nucleotídeos: Adeína (A), Citocina (C), Guanina (G) e Timina (T). Por outro lado, o RNA é uma cadeia de quatro nucleotídeos, tendo Uracila (U) em vez da Timina (T). O processo de sequenciamento é a decodificação da sequência de nucleotídeos dos cromossomos de um organismo. As sequências de RNA são gerados a partir de regiões particulares de DNA, formando dessa forma transcritos que codificarão proteínas, onde a coleção desses transcritos é chamada de transcritoma.

As tecnologias de sequenciamento de alto desempenho geram quantidades massivas de fragmentos de DNA/RNA. O comprimento desses fragmentos é muito pequeno quando comparado com os tamanhos DNA/RNA completos. Os *pipelines* computacionais devem reconstruir a molécula de DNA/RNA inteira a partir dos fragmentos sequenciados chamados SRS (*short read sequences*). Desde que uma SRS apresenta baixo significado biológico de forma desassociada, diferentes análises devem ser feitas para extrair informação biológica relevante das SRS.

Neste trabalho as SRS são fragmentos de RNA gerados pelos sequenciadores de alto desempenho. Após serem geradas, as SRS passam por múltiplas análises, tais como: (i) avaliar a qualidade dos dados; (ii) filtrar erros de sequenciamento; (iii) armazenar sequências de banco de dados externos ao laboratório de bioinformática; (iv) buscar funcionalidades biológicas e (v) armazenar resultados produzidos pelos diferentes sistemas usados. Como dito antes, essas análises geram grandes quantidades de dados, então é

essencial criar modelos de dados que representem, organizem e garantam o acesso aos dados biológicos nas diferentes fases do *pipeline* de sequenciamento de alto desempenho.

Neste contexto, o uso de um SGBD (Sistemas de Gerenciamento de Banco de Dados) ou sistema de arquivos tem um papel crucial para resolver os desafios de armazenamento e administração de grande volume de dados que têm características peculiares como os dados biológicos. Além desse problema, esses dados precisam de modelos adequados que representem a informação gerada nos laboratórios de biologia molecular. Sistemas de *pipelines* têm sido criados para lidar com as diferentes fases de um projeto de sequenciamento de genoma. A dificuldade na organização e o armazenamento dos dados gerados pelas diferentes fases e programas de um *pipeline* de sequenciamento de alto desempenho é o foco do nosso trabalho.

Particularmente, o interesse no uso de um SGBD dar-se-á pelas vantagens que ele fornece, tais como: segurança, organização, fácil consulta aos dados e compressão dos dados, o que é muito útil para administrar dados biológicos. Neste trabalho, usamos a SGBD relacional para armazenar dados ao longo das diferentes fases do *pipeline* de sequenciamento de alto desempenho. O SGBD relacional traz vantagens: amplo uso no mercado, muitos SGBDs relacionais comerciais e de código aberto são disponibilizados, a existência de padrões e a facilidade de uso da linguagem de consulta.

1.1 Motivação

Os *pipelines* para projetos de sequenciamento de genomas são implementados como meio para administrar, especificar e coordenar a execução de experimentos que envolvam diferentes fases com características particulares e com fins específicos. Eles permitem a execução de tarefas que usam dados e ferramentas heterogêneos. Há diversos sistemas para gerenciar experiências da Bioinformática. Esses sistemas cumprem seus propósitos fornecendo ferramentas para as diferentes fases do *pipeline*.

Na maioria dos *pipelines* usados, o foco está na utilização das ferramentas, e não há uma preocupação real de como os dados gerados em cada fase serão armazenados, organizados e gerenciados. Como consequência, tem-se dados pouco organizados, altamente redundantes dificultando o desenvolvimento das análises sobre os dados gerados. Em relação a esses aspectos, é importante ter modelos conceituais que possam representar os dados e processos de forma adequada de um projeto de sequenciamento de alto desempenho. Sendo assim a pesquisa na área de modelos de dados biológicos aplicados às fases de um *pipeline* de sequenciamento de alto desempenho transcritoma, será importante para os projetos de sequenciamento atuais e futuros.

1.2 Objetivos

Esta dissertação tem como objetivo geral o desenvolvimento de modelos de dados para um *pipeline* de sequenciamento de alto desempenho, onde seja possível a representação dos dados das diferentes fases desse *pipeline*. Esses modelos de dados, referem-se a representação conceitual que utiliza a abordagem orientada a objetos, assim como também a implementação do modelo através de um esquema relacional.

1.2.1 Objetivos Específicos

No intuito de atingir o objetivo geral desta dissertação, foram definidos alguns objetivos específicos:

1. Definir o *pipeline* de sequenciamento de alto desempenho a ser usado;
2. Desenvolver um modelo de dados conceitual envolvendo as diferentes etapas do *pipeline* de sequenciamento de alto desempenho definido;
3. Desenvolver um esquema relacional para o modelo de dados definido;
4. Implementar o esquema relacional em um sistema gerenciador de banco de dados relacional.
5. Desenvolver estudos de caso com dados reais para validar o modelo desenvolvido.

1.3 Estrutura do Trabalho

A estrutura desse trabalho é apresentada a seguir.

No Capítulo 2, são apresantados os conceitos básicos de Biologia Molecular e Bioinformática, especialmente o projeto de um *pipeline*, necessário ao entendimento do trabalho.

No Capítulo 3, discutimos modelagem de dados, em particular modelos para bioinformática.

No Capítulo 4, propomos um modelo de dados para um *pipeline* de sequenciamento de alto desempenho.

No Capítulo 5, discutimos dois estudos de caso onde o modelo proposto é implementado e os resultados práticos são discutidos.

Finalmente, no Capítulo 6 concluimos e sugerimos trabalhos futuros.

Capítulo 2

Conceitos Básicos de Biologia Molecular e Bioinformática

O presente capítulo apresenta conceitos fundamentais de Biologia Molecular e Bioinformática, necessários ao entendimento deste trabalho. A Seção 2.1 apresenta de forma breve conceitos de proteínas e ácidos nucleicos (DNA e RNA). Além disso, é exposto o dogma central da biologia molecular, ou o processo através do qual as informações contidas no DNA são utilizadas para a síntese de proteínas. Na seção 2.2, apresentamos conceitos de bioinformática, mas particularmente, falamos sobre o sequenciamento de alto desempenho do Illumina, transcritomas, as fases de um *pipeline* para projetos transcritomas de alto desempenho, e bancos de dados biológicos.

2.1 Biologia Molecular

A Biologia Molecular é o ramo da Biologia responsável pelo estudo da estrutura de proteínas e ácidos nucléicos, processos e outros atores envolvidos como organelas celulares e enzimas [4].

Segundo a ciência moderna, a vida originou-se há 3,5 bilhões de anos [4] com formas de vidas muito simples, mas com o percorrer do tempo, estes organismos foram mudando sua aparência e suas estruturas biológicas por um processo denominado evolução para depois dar origem aos organismos mais complexos, pluricelulares, convivendo com organismos mais simples, unicelulares, como os procariotos.

Partindo do mesmo ponto inicial – estas formas de vida primordiais – pode-se compreender que todos os organismos, unicelulares ou pluricelulares, dividem uma composição parecida. A composição química das células de organismos vivos é predominantemente formada por carbono (C), oxigênio (O), nitrogênio (N) e hidrogênio (H).

Os organismos simples, assim como os complexos, possuem uma química molecular básica, onde os compostos aparecem, muitas vezes, como cadeias de outros pequenos compostos interligados. Estes pequenos compostos são os monômeros e as cadeias formadas pela união repetida de monômeros são os polímeros. Os polímeros mais importantes para os organismos são as proteínas e os ácidos nucléicos, formados respectivamente por cadeias de aminoácidos e cadeias de base nitrogenadas. De forma geral, as proteínas são as responsáveis, pelo que um ser vivo é e faz em um sentido físico funcional. Por outro lado, os ácidos nucleicos são encarregados de codificar essa informação para produzir pro-

teínas e como mecanismos de armazenamento para preservar a continuidade dos diferentes organismos.

2.1.1 Proteína

Os organismos da natureza na sua maioria são feitos de proteínas que são cadeias de moléculas chamadas aminoácidos. Um aminoácido tem sua estrutura química geral representada pela Figura 2.1. Na natureza existem muitos aminoácidos, mas no corpo humano só são usados 20 aminoácidos que são denominados primários.

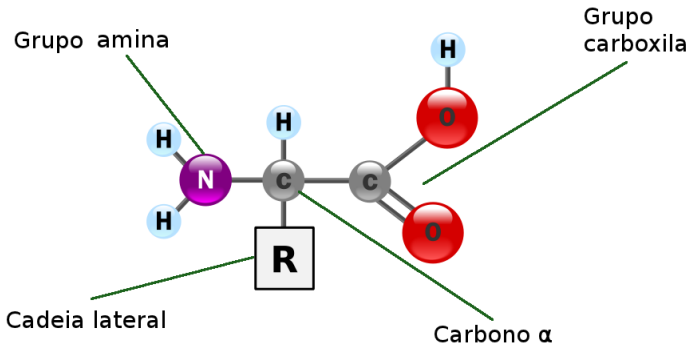

Figura 2.1: Estrutura geral dos aminoácidos adaptado de [1].

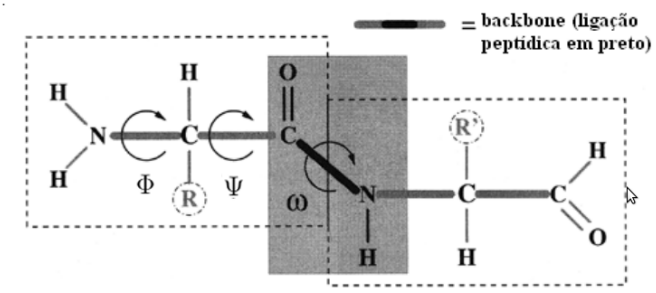

Figura 2.2: Ligação peptídica e orientações Φ e Ψ do carbono Cα [2].

Composição Química

Proteínas, como ditas anteriormente, são formadas por longas cadeias de aminoácidos. A estrutura do monômero aminoácido é formada essencialmente por um carbono central, um grupo amina, um grupo carboxila e uma cadeia complementar (radical), que é responsável pela existência de vários aminoácidos únicos na natureza [4]. A Figura 2.1 mostra os grupos formadores. Somente aminoácidos entram na composição de proteínas [12].

Para a formação das proteínas, dois aminoácidos são combinados através de uma reação de sínteses por desidratação (gera uma molécula de água por cada união), onde o carbono do grupo carboxila de um aminoácido liga-se ao átomo de nitrogênio do grupo amina do outro aminoácido, gerando uma molécula de água neste processo, esse tipo de ligação é chamada de ligação peptídica, criando resíduos de aminoácidos que formaram as proteínas.

A proteína formada pelo encadeamento de aminoácidos é uma sequência linear, conhecida de estrutura primária, mas a proteína tem estrutura secundária, terciária e quaternária, que formam a estrutura tridimensional da proteína (Figura 2.3). Na natureza, são catalogados 22 aminoácidos conhecidos [12], sendo 20 os mais comumentes achados em proteínas, e 2 aminoácidos raramente encontrados em polipeptídeos, eles são listados na Tabela 2.1.

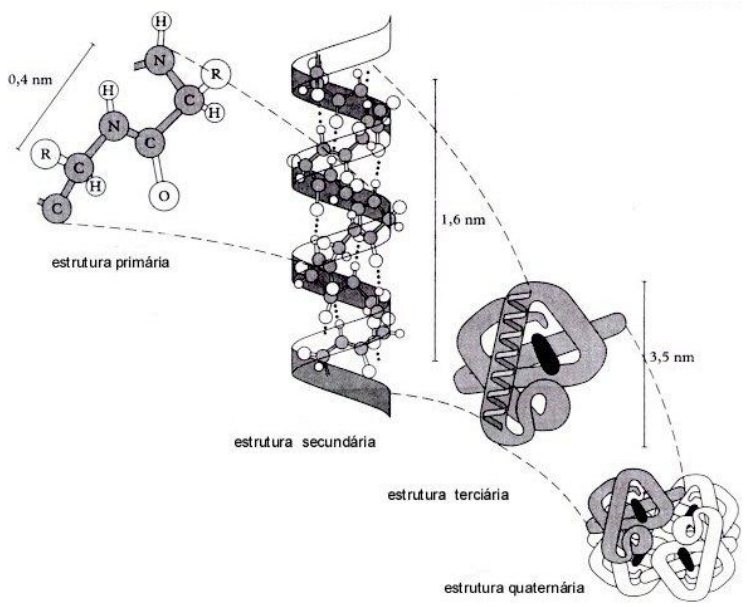

Figura 2.3: Estrutura primária, secundária, terciária e quaternária da molécula da hemoglobina [3].

Estrutura da Proteína

A função de uma proteína é determinada pela sua estrutura espacial [17]. Os peptídeos que a compõe combinam-se por meio de ligações de hidrogênio (chamadas também de ponte de hidrogênio), ligações iônicas e ligações dissulfúricas (entre átomos de enxofre dos resíduos de aminoácidos Cisteina (Cys)). Outros determinantes da conformação espacial

Tabela 2.1: Lista dos 22 aminoácidos encontrados na natureza [12]. Os aminoácidos marcados com (*) são aminoácidos raramente encontrados.

	Nome	**Abreviação**	**Código**
1	Alanima	Ala	A
2	Arginima	Arg	R
3	Asparagina	Asn	N
4	Ácido Aspártico	Asp	D
5	Asparagina ou Ácido Aspártico *	Asx	B
6	Cisteína	Cys	C
7	Glutamina	Gln	Q
8	Ácido glutâmico	Glu	E
9	Glutamina ou Ácido glutâmico *	Glx	Z
10	Glicina	Gly	G
11	Histidina	His	H
12	Isoleucina	Ile	I
13	Leucina	Leu	L
14	Lisina	Lys	K
15	Metionina	Met	M
16	Fenilalanina	Phe	F
17	Prolina	Pro	P
18	Serina	Ser	S
19	Treonina	Thr	T
20	Triptofano	Trp	W
21	Tirosina	Tyr	Y
22	Valina	Val	V

de proteínas são a hidrofobicidade de regiões de polipeptídeo – isto é, o grau de afinidade com moléculas de água – e a rotação dos eixos Φ e Ψ (Figura 2.2).

A sequência de resíduos que forma a proteína é dita estrutura primária da mesma, chamada também de *estrutura linear*. Esta estrutura é importante para a leitura dos aminoácidos que compõem a proteína, porém não caracteriza sua função [4].

A *estrutura secundária* da proteína está composta pelos alinhamentos e dobramentos das estruturas lineares, principalmente pelos dobramentos nos eixos ϕ e ψ, e pelo alinhamento de *backbones* (Figura 2.2) formando assim as tão conhecidas cilíndricas *α-hélice*. Também é possível encontrar alinhamentos do tipo *β-folha*, onde o esqueleto polipeptídico está quase completamente estendido. As repetições de padrões de alinhamento e dobramento da estrutura linear nesta configuração espacial são chamadas de motivos. Motivos são importantes para inferência de funções e grau de similaridade entre diferentes proteínas [17].

A *estrutura terciária* resulta do enrolamento da *α-hélice* ou da *β-folha* descrevendo o dobramento final de uma cadeia, enquanto a estrutura secundária é determinada pelo relacionamento estrutural de curta distância, a terciária é caracterizada pelas interações de longa distância entre aminoácidos. Finalmente, a estrutura quaternária da proteína

considera sua totalidade em forma tridimensional, o que significa que é a forma natural como é encontrada no organismo.

2.1.2 Ácidos Nucléicos

Segundo a biologia contemporânea, os ácidos nucléicos tem a função principal de armazenar informação necessária para criação de proteínas e possibilitar a transferência desta informação para gerações futuras desses organismos, através de processos de reprodução celular [4]. Os seres vivos têm dois tipos de ácidos nucleicos: DNA - ácido desoxirribonucleico - e RNA - ácido ribonucleico - são ambos os polímeros compostos de moléculas mais simples - monômeros - os *nucleotídeos*. No caso de DNA e RNA, tem-se um grupo fosfato, um açúcar central e uma base nitrogenada, formando um *nucleotídeo* [17, 4]. A composição em cadeias de nucleotídeos forma uma sequência de DNA ou RNA, dependendo da composição destes nucleotídeos.

DNA

O DNA é um ácido nucleico formado por um açúcar central - a pentose (açúcar com cinco átomos de carbonos) desoxirribose (Figura 2.4) - e uma base nitrogenada - moléculas com ciclos de carbonos e nitrogênios- (Figura 2.5).

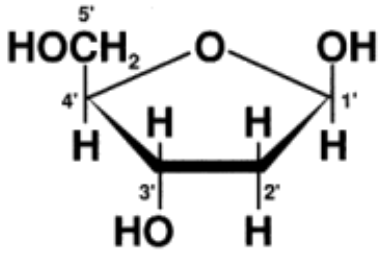

Figura 2.4: Açúcar pentose principal que compõe o nucleotídeo criador do DNA: a desoxirribose.

Na Figura 2.4 tem-se os carbonos numerados de 1' a 5' por convenção em relação à estrutura química do composto. O carbono 1' está associado a uma base nitrogenada, ao carbono 5' o fosfato. No carbono 3' ocorre a reação da ligação entre o fosfato do nucleotídeo com o grupo hidroxila do carbono 3' do nucleotídeo ao qual está se ligando. Naturalmente, por causa desta ligação, a molécula de DNA é orientada do carbono 5' ao carbono 3' [17].

A molécula de DNA é de dupla fita. As duas fitas formam uma estrutura de hélice (Figura 2.6), sendo descobertas por James Watson e Francis Crick em 1953. A dupla fita se mantém dessa forma graças às uniões entre duas bases, cada uma de uma fita diferente, essa união acontece graças à natureza complementar delas, neste caso as púricas (Adeína e Timina) unem-se com as pirimidinas (Citosina e Timina) [13] (Figura 2.5) . Isto acontece por causa da afinidade eletrônica da molécula. Pela natureza complementar das bases, é possível extrair o complemento de uma fita do DNA aplicando a seguinte regra:

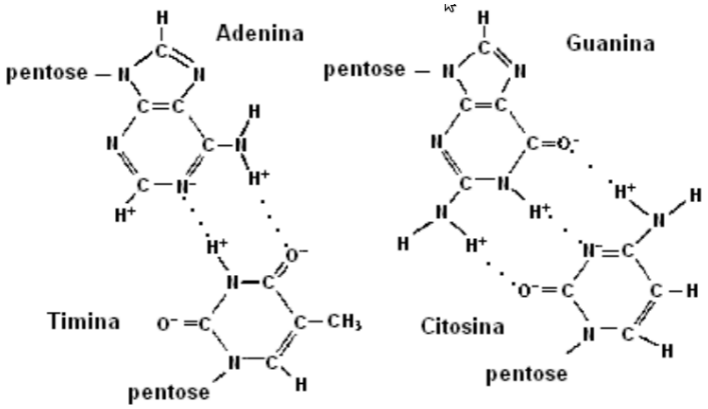

Figura 2.5: Bases nitrogenadas que compõem um nucleotídeo da molécula DNA adaptado de [4].

$$\text{Adenina} \longleftrightarrow \text{Timina}$$
$$\text{Guanina} \longleftrightarrow \text{Citosina}$$

onde a $\longleftrightarrow$, representa o complemento de uma base pode ser feita em ambos sentidos.

A disposição espacial de uma fita de DNA, indo de 5' ao 3', pode- se concluir que seu complemento é exato oposto, indo de 3' ao 5'. Portanto, uma fita é o exato complemento reverso da outra, dando origem à duplicação de trechos do código de DNA.

Grande parte do material genético encontrado no DNA de organismos eucariotos não codifica para proteínas [18]. Denomina-se genes as regiões delimitadas do DNA que codificam para proteínas ou RNAs [4], isso no ato de transcrição, o DNA é transcrito para um RNA funcional válido ou para um RNA mensageiro válido (veja seção 2.1.3).

RNA

O RNA é um aminoácido como o DNA com certas diferenças e funcionalidades específicas. O RNA é formado pelo açúcar ribose (veja Figura 2.7), a diferença do DNA que tem a 2'- desoxirribose. Esta molécula está composta por uma base nitrogenada além das já descritas, ela é a Uracila (U) que substitui a Timina (T). Outra diferença com o DNA é que só tem uma cadeia ou fita única de nucleotídeos (ver Figura 2.6), tendo diferentes formatos de acordo com a função que pode exercer.

Identicamente ao DNA, a orientação do RNA se dá do carbono 5' ao carbono 3'. Podem-se reescrever as regras de complementaridade das bases nitrogenadas simplesmente trocando-se a base Timina pela base Uracila.

A estrutura de fita única faz ao RNA vulnerável a danos e erros, portanto menos apto a transportar informação genética [17]. Por essa característica, além da estrutura química mais simplificada tanto da base Uracila (Figura 2.8), como da estrutura do RNA, existem teorias de que o RNA teria sido o primeiro ácido nucléico a ser usado como transportador de material genético [19].

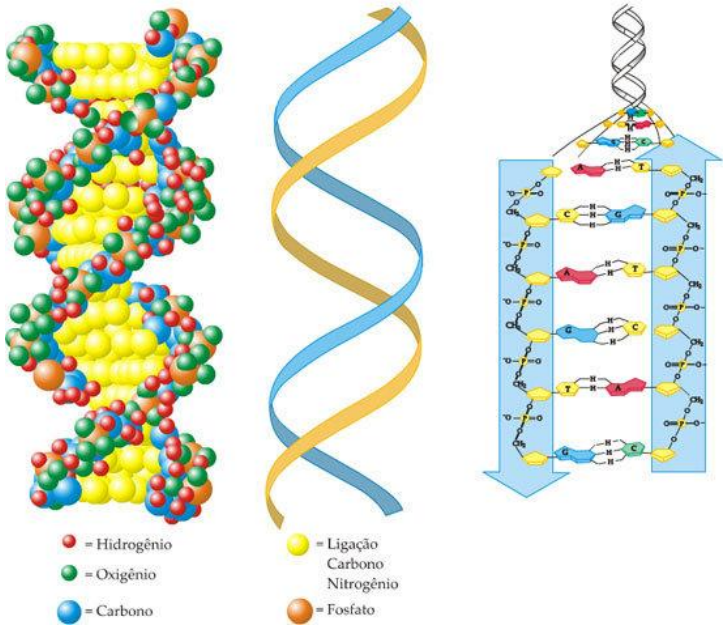

Figura 2.6: A dupla Hélice do DNA mostrando a união das bases [1].

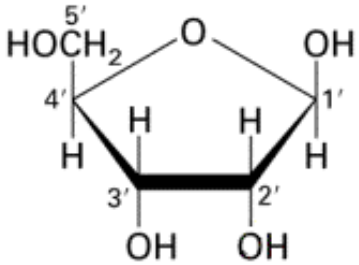

Figura 2.7: Açúcar principal do nucleotídeo formador do RNA: a ribose adaptado de [4].

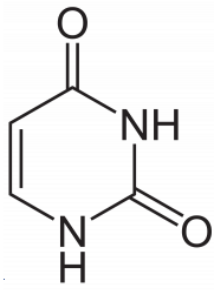

Figura 2.8: Uracila - base pirimidina que compõe um nucleotídeo de molécula RNA.

2.1.3 Dogma Central da Biologia Molecular

O dogma central define o paradigma da Biologia Molecular, no qual a informação é perpetuada através da replicação do DNA e é traduzida através dos processos de transcrição e tradução. A transcrição que converte a informação do DNA em uma forma mais acessível (uma fita de RNA complementar) e através da tradução, o código genético contido no RNA é traduzido em proteínas (Figura 2.9).

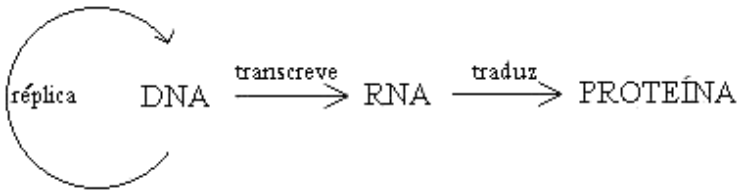

Figura 2.9: Dogma central da Biologia.

A replicação é o processo pelo qual é gerada uma cópia idêntica a uma molécula de DNA, envolvendo um conjunto de proteínas. Por outro lado, a transcrição começa reconhecendo o início de um gene graças a uma pequena região do DNA sinalizado como o começo de um gene, chamado promotor. Tendo localizado o gene, a célula copia a informação do gene criando uma molécula de RNA complementar a uma fita de DNA. Esta molécula de RNA é chamada de RNA mensageiro ou mRNA (messenger RNA). Assim o mRNA possui a mesma sequência de uma das fitas de DNA, contudo tendo a base U no lugar da T. Este processo é chamado de transcrição. O processo de transcrição descrito acima é válido para seres chamados de procariotos, organismos sem núcleo celular e com o DNA flutuando livremente na célula. Já em organismos chamados eucariotos, seres onde o DNA está armazenado em um núcleo celular, o processo de transcrição é um pouco mais complexo. Os genes dos seres eucariotos são compostos de duas partes, os intros e os exons. Após a transcrição, os introns são removidos do mRNA. Sendo assim, em um organismo eucarioto nem todas as bases de um gene são utilizadas na transcrição. Ao DNA contendo todas as bases do gene denominamos DNA genômico e às bases do DNA presentes no mRNA após a remoção dos introns chamamos cDNA(DNA codificador).

Feita a transcrição, o processo de tradução começa nos ribossomos onde a proteína será sintetizada. Os ribossomos são estruturas compostas de proteínas, tipos especiais de RNA, chamados de RNA ribossômico e abreviado como rRNA. Os ribossomos funcionam como linhas de montagem de proteínas, lendo a informação para a síntese do mRNA e utilizando moléculas conhecidas como tRNA(transfer RNA) para realizar a tradução dos códons para os aminoácidos correspondentes. Os mecanismos celulares realizam a junção de diversos aminoácidos. Mais detalhadamente, os RNAs são as moléculas responsáveis por efetuar a conexão entre os códons e os aminoácidos correspondentes. Cada tRNA é composto de duas partes, uma delas possui afinidade química a um dado códon, enquanto a outra se liga com facilidade ao aminoácido correspondente ao códon. Conforme a fita de mRNA passa pelo ribossomo, um tRNA correspondente ao códon sendo lido pelo ribossomo liga-se ao mesmo em questão trazendo consigo o aminoácido correspondente. Uma enzima então catalisa a ligação peptídica para adicionar o aminoácido em questão a proteína. A síntese prossegue assim, um aminoácido de cada vez, parando apenas quando

um códon do tipo STOP é encontrado. Quando isso ocorre, a proteína desliga-se do ribossomo, é liberada na célula. O mRNA é degradado para posterior reaproveitamento dos seus componentes.

2.2 Bioinformática

A Bioinformática é uma área multidiciplinar que envolve ciências como Biologia Molecular, Estatística, Matemática e Ciência da Computação, e tem como objetivo realizar análises de dados biológicos, como sequências de bases de DNA e genes, predizer a estrutura e função de diversas macromoléculas [20]. A Bioinformática surgiu quando foi necessário o uso de ferramentas computacionais para análises de dados genéticos, originado com os projetos genoma, na década de 1990. Portanto, é um ramo do conhecimento relativamente recente.

A bioinformática enfatiza o desenvolvimento de ferramentas para realizar o armazenamento e manipulação dos dados biológicos gerados durante um projeto de sequenciamento. Com o atual volume de dados produzidos pelos projetos de sequenciamento, a utilização de ferramentas computacionais traz grandes auxílios aos biólogos, ao permitir a recuperação rápida dos dados armazenados de um projeto genoma e apresentar os resultados de maneira a facilitar a análise dos mesmos e assim auxiliar na descoberta de funções para as sequências obtidas.

2.2.1 Tecnologias de Sequenciamento de Alto Desempenho

Embora o sequenciamento Sanger [21] tenha sido a técnica de sequenciamento mais usada durante os últimos anos, novas técnicas de sequenciamento massivamente paralelos são usadas atualmente nos projetos de sequenciamento, revolucionando a forma como se realiza o sequenciamento de DNA no mundo. A grande demanda por sequenciamento de baixo custo têm estimulado o desenvolvimento de tecnologias de sequenciamento massivamente paralelos, produzindo milhões de sequências em uma só rodada [22]. Nesse sentido, as tecnologias de sequenciamento massivamente paralelo são destinadas a baixar o custo de sequenciamento de DNA o máximo possível.

Nesse contexto, uma variedade de sequenciadores de alto desempenho produzem um número muito grande de sequências de DNA. Hoje em dia, os sequenciadores massivamente paralelos que estão disponíveis comerciavelmente são: Pirosequenciamento 454 [16], realizada sobre o Sequenciador Genômico FLX, o qual foi disponibilizado pelas companhias 454 *Life Science* e Roche Applied Science (http://www.454.com); Illumina [16] com o GAII (Genome Analyzer II); tecnologias SOLiD que usa uma técnicas de sequenciamento por ligação. Recentemente, outras duas tecnologias foram anunciadas: O Helicos Heliscope e Pacific Biosciences SMRT [16].

Nos estudos de caso desta dissertação foram utilizados dados gerados pelo sequenciador Illumina. Este sequenciador foi desenvolvido pela Solexa, subsequentemente adquirido pela Illumina. Atualmente, os Sequenciadores Illumina GAI e GAII fazem uso do sequenciamento por sínteses. O sequenciador GA I produz SRS de 25 a 35 pares de bases [23] e o sequenciador GA II produz aproximadamente 50 pares de bases [16] com mais de 2000

Mb de dados por corrida ao longo de aproximadamente quatro dias. O método de Illumina é um dos mais amplamente usados em sua curta existência; aplicações publicadas incluem expressão de genes, descoberta de SNP, resequenciamento, e experimentos ChIP (Chromatin Immunoprecipitation).

2.2.2 Projetos Transcritoma

O transcritoma é o conjunto completo de transcritos de uma célula, e seu estudo é realizado em um estado específico do seu desenvolvimento ou condição fisiológica. Entende-se que um transcritoma é essencialmente o meio para interpretar o elemento funcional do genoma e revelar os constituintes moleculares das células e tecidos. O transcritoma tem muitos objetivos, entre os quais: catalogar todos os tipos de transcritos, incluindo o mRNA, ncRNA (non-coding RNA) e pequenos RNAs; determinar a estrutura transcricional dos genes; quantificar os níveis expressão de cada transcrito durante o desenvolvimento da celula e baixo diferentes condições; entre outras.

Como visto na Seção 2.1.1, a síntese de uma proteína ocorre através da transcrição das informações contidas no DNA em um RNA mensageiro e posterior tradução desta informação em aminoácidos. Dizemos então que o gene codificando a proteína em questão é expresso. O conjunto dos RNAs mensageiros de uma célula é chamado de transcritoma, e projetos de sequenciamento visando a obtenção desses RNAs mensageiros são conhecidos como projetos transcritoma.

A obtenção de todos os transcritos de uma determinada célula de um dado organismo é uma tarefa complexa, pois nem todos os genes são expressos a todo momento. De fato, durante diferentes fases da vida de um organismo, diferentes genes são expressos em diferentes intensidades. Dessa forma, grande parte dos projetos transcritoma envolvem o sequenciamento dos RNAs mensageiros em um dado estado da vida do organismo de interesse, podendo este ser durante o desenvolvimento de uma planta, a metamorfose de um inseto ou mesmo a ocorrência de um câncer. Uma das principais informações obtidas através dos transcritomas é o conjunto de genes expressos durante uma dada condição de um organismo, por exemplo, durante uma infecção.

Para a obtenção dos transcritomas, uma técnica muito utilizada consiste em capturar os RNAs mensageiros de uma célula exposta a dadas condições, e a partir da mesma gerar a sequência de DNA cuja transcrição originou o mRNA. Conforme exposto na Seção 2.2.1, essa fita de mRNA é complementar a sequência de DNA que a originou. Portanto, para obter a sequência de nucleotídeos efetivamente expressos durante a produção da proteína em questão, basta obter o complemento desta fita de RNA. A sequência de DNA obtida desta maneira é conhecida como DNA codificador ou cDNA. Para determinar o transcritoma, procede-se ao sequenciamento dos cDNAs, seja através do método Sanger [21], ou por meio dos novos sequenciadores de alto desempenho.

2.2.3 *Pipelines* para Projetos Transcritoma

Pipelines são sistemas computacionais que executam sequencialmente uma série de programas, onde os resultados de um programa são usados como entrada do próximo programa na linha de execução [24]. Tradicionalmente, um projeto de sequenciamento (Sanger) tem três fases importantes: submissão, montagem e anotação [21], mas de-

vido às diferentes caraterísticas das sequências obtidas pelos novos sequenciadores, novos *pipelines* devem ser implementados. Neste contexto, são desenvolvidos os *pipelines* de sequenciamento de alto desempenho para superar as limitações dos *pipelines* tradicionais. A definição do *pipeline* de sequenciamento de alto desempenho dependerá, entre outros, do sequenciador e das caraterísticas dos dados gerados por esse sequenciador, podendo contemplar três fases principais: filtragem, mapeamento e análise.

A fase de *filtragem* começa após serem sequenciadas as amostras de DNA/RNA pelo sequenciador de alto desempenho. Os métodos aplicados variam dependendo da tecnologia usada, mas os resultados do sequenciamento são sequências de caracteres sendo armazenadas em formatos adequados para serem usados nos processamentos computacionais das fases posteriores. Geralmente, os arquivos resultantes são de formato texto que contém as sequências de bases identificadas e as qualidades associadas a cada base. De forma geral, o valor da qualidade é a probabilidade de erro na identificação de uma determinada base. O processo de sequenciamento pode conter erros originados pela presença de contaminantes que afeta a qualidade dos resultados, ou simplesmente erro de sequenciamento. Alem disso, tem-se regiões sequenciadas que não interessam ou que podem dificultar o processamento nas próximas etapas do *pipeline*. No intuito de conseguir resultados confiáveis, essas regiões devem ser removidas. O objetivo desta fase é a remoção das sequências que possam dificultar e afetar negativamente (erros e resultados pouco confiáveis) nos resultados das próximas fases. Nesse contexto, nesta fase são removidos fragmentos tais como *primers*, vetores, adaptadores, e longas sequências de bases repetidas que simplesmente não são de interesse ou que de alguma forma possam afetar nas análises das próximas fases [25].

Após a fase de filtragem começa a fase de *mapeamento*, onde são usados um ou mais programas para que os diferentes fragmentos de DNA ou cDNA (mRNA, cujos íntrons já foram removidos) que tenham qualidade desejada sejam localizados dentro de um genoma de referência de um organismos próximo ao organismo estudado. A solução de um quebra-cabeça é uma analogia ao processo de mapeamento onde as SRS seriam as peças do quebra-cabeça e o genoma de referência seria o quebra-cabeça todo. Neste sentido, a solução desse quebra-cabeça é procurar onde poderiam encaixar as SRS dentro desse enorme quebra-cabeça chamado genoma de referência. A procura das localizações mais adequadas das SRS são feitas por comparação, em particular observando sobreposições no genoma de referência [4].

O processo de mapeamento é muito importante, já que é uma fase que ajuda a encontrar genes, particularmente aqueles envolvidos em doenças humanas. Por exemplo, os pesquisadores estudam famílias inteiras afetadas por uma doença, seguem o rastro de doenças hereditárias por muitas gerações. Regiões, que tendem a ser herdadas junto com a doença tendem a ser localizados próximos ao gene da doença e torna-se "marcadores" para o gene em questão [26].

A fase de *análise* constitui a última fase do *pipeline* de sequenciamento. Nesta fase os pesquisadores procuram identificar os genes presentes nas regiões mapeadas na fase anterior e também outras informações como as funções biológicas, participação em vias metabólicas e relações filogenéticas desses genes, entre outras importantes funções. Portanto, a fase de análise é um processo de interpretação dos dados brutos gerados pelo sequenciamento com o objetivo de acrescentar informações biológicas. A fase de análise é realizada por sistemas computacionais que tentam inferir as funções biológicas das

sequências de DNA. O processo de identificação de genes é feito através de comparações das SRS mapeadas com genes já conhecidos, cujas sequências de nucleotídeos estão disponíveis em banco públicos. Os resultados obtidos são analisados pelos biólogos que podem confirmar, mudar ou recusar as sugestões das análises feitas. As sugestões também podem ser utilizadas para a realização de experimentos significativos ao trabalho de pesquisa do organismo.

O *pipeline* descrito aqui é genérico mas, adaptável a uma série de projetos com diferentes objetivos e técnicas. Cabe notar em geral que o processamento realizado em cada etapa é dividido em uma série de programas de Biologia Computacional. A correta integração desses programas no *pipeline* auxilia e acelera o processo de análise assim como a automatização das etapas. A adequação dos *pipelines* é consequência da configuração dos parâmetros em cada fase para atingir os objetivos dos projetos.

2.2.4 Bancos de Dados Biológicos

Os banco de dados biológicos são importantes principalmente para proporcionar à comunidade científica uma forma de tornar os dados acessíveis de forma fácil e rápida. Com o sequenciamento de larga escala, foi necessário a construção de bancos de dados mais robustos para armazenar o grande volume de sequências (DNA, RNA e proteínas) obtidas pelos pesquisadores, entre os bancos de dados mais usados temos: EMBL, NCBI - GenBank, COG, KEGG, SWISS-PRO, TrEMBL e o RefSeq. Cada um desses possibilita a submissão individual de sequências de DNA e trocam informações entre si diariamente, sendo que todos eles atualizam diariamente as sequências disponíveis para os pesquisadores [27].

O **GenBank** é um banco de dados que contém sequências de DNA disponíveis publicamente para mais de 165.000 organismos conhecidos, obtidas principalmente através da submissão de laboratórios individuais e de lotes de submissão de projetos de sequenciamento em larga escala [28]. Em mais de 20 anos desde seu estabelecimento, GenBank tem-se convertido em um banco de dados muito importante e influente para a pesquisa nos diferentes campos da biologia. A taxa de crescimento exponencial dos dados do GenBank continua desde sua fundação e cada 18 meses são dobrados seus dados [28].

Cada registro no GenBank consiste em uma sequência e sua anotação, que são associados a um identificador único, o número de acesso, que permanece constante durante a existência do registro, mesmo quando há uma mudança em sua anotação.

O banco de dados de sequências de nucleotídeo **EMBL** (*European Molecular Biology Laboratory* ou conhecido como EMBL-Bank) é a atividade central do EBI (Instituto Europeu de Bioinformatica). O banco de dados EMBL coleta, organiza e distribui um banco de dados de sequências de nucleotídeos e informação biológica relacionadas [27].

O **PDB** (*Protein Data Bank*) é um repositório para dados estruturais 3-D de grandes moléculas biológicas, tais como proteínas e ácidos nucleicos. O PDB é a fonte importante para áreas de biologia estrutural, tais como genômica estrutural [29].

O **COG** (*Clusters of Orthologous Groups*) constituído por grupos ortólogos de proteínas (produzidas por genes derivados de um ancestral comum que se diferenciou devido a divergências dos organismos associados a eles; tais genes tendem a ter funções semelhantes). Cada COG representa uma função genômica conservada durante o processo evolutivo, ou seja, funções que se desenvolveram desde cedo e se mantiveram nas espécies

atuais. Parte-se do pressuposto que sequências antigas que se conservaram ao longo do tempo formam um núcleo mínimo de funcionalidades exigido por uma espécie moderna. Para integrar o banco, é necessário que o COG esteja presente em pelo menos três linhagens de organismos. Consultas ao banco podem ser feitas, por exemplo, através da categoria funcional e do padrão filogenético [30].

O **KEGG** (*Kyoto Encyclopedia of Genes and Genomes*) é um banco de dados que utiliza conhecimentos de interações moleculares, de genes, proteínas e de compostos químico e suas reações para identificar um produto genômico dentro das vias metabólicas existentes neste banco [31].

O **SWISS-PROT** é um banco de dados secundário que consiste apenas de sequências de proteínas. Para cada sequência no banco de dados tem-se dados da molécula em questão e anotação biológica da mesma. A anotação da proteína é bastante completa abarcando várias caraterísticas onde a ideia é adicionar o maior número possível de informações relativas aquela proteína no Swiss-Prot. E assim como o RefSeq, o Swiss-Prot também tem a intenção de produzir a menor redundância possível em relação às entradas de proteínas presentes no banco. Além disso, Swiss-Prot apresenta referência cruzadas com outras bases de dados de biomoléculas, dessa forma facilitando a apressão de informação sobre a sequência de proteínas em questão [32].

O **TrEMBL** é o complemento do SWISS-PROT que contém as traduções das CDS (sequências codificantes) presentes no banco de sequências EMBL, ainda não integradas ao SWISS-PROT [27, 32].

O **RefSeq** (*Reference Sequence*) tem como objetivo produzir um conjunto não redundante de sequências de DNA genômico, transcritos (cDNA) e de proteínas de diferentes organismos. Ele é resultado da curadoria manual realizado pelo NCBI (*National Center for Biotechnology Information*), ou seja, pesquisadores treinados analisam sequência por sequência e as informações relevantes são adicionadas ao banco de dados RefSeq. Uma das características mais interessantes do RefSeq é ser capaz de reunir vários dados divergentes em uma plataforma consistente e apresenta um conjunto de padrões e convenções comuns [33].

Capítulo 3

Modelos de Dados para Bioinformática

No presente capitulo são apresentados os conceitos teóricos fundamentais de modelagem de dados e o estudo desse tema na área da bioinformática. Na seção 3.1, o foco principal é na modelagem de dados de maneira geral, abordando as principais características de um modelo de dados, assim como, a sua importância em um sistema computacional. Nessa seção também são apresentados, os modelos de dados mais usados na atualidade para representar um conjunto de requerimentos dos sistemas. Na seção 3.2, é apresentado o estado da arte dos trabalhos relacionados à modelagem de dados da bioinformática. No final da Seção 3.2 uma análise comparativa entre os diferentes modelos é realizada.

3.1 Modelagem de Dados

A modelagem de dados é uma das etapas mais importantes de um projeto de sistemas de informação, pois a escolha de um modelo que se ajuste à realidade que se pretende representar é um fator crítico para conseguir ótimos resultados nos sistemas desenvolvidos [34]. Através da modelagem de dados é especificado um modelo de dados. Segundo [35] a modelagem de dados é uma coleção de ferramentas conceituais para descrever o relacionamento, a semântica e as restrições dos dados.

De maneira resumida, a modelagem de dados é uma maneira de expressar a realidade de forma abstrata usando um formalismo, e existem diversas técnicas de modelagem de dados, que se adaptam para representar uma realidade em particular. A modelagem de dados, pode ser usada para especificação das regras de um negócio, assim como também, para estruturar um banco de dados, por exemplo. Ela faz parte do ciclo de desenvolvimento de um sistema de informação que é de vital importância para o correto resultado do projeto. O processo de modelar dados tem como objetivo desenhar sistemas (comumente usado para sistemas de informação), observando com atenção o papel dos componentes desse sistema, as dependências lógicas e a relação estabelecida entre esses componentes. Desta forma, pode-se concluir que o método de modelar dados consiste em uma série de aplicações teóricas e práticas, com o objetivo de construir um modelo de dados consistente não redundante aplicável em um sistema de banco de dados.

3.1.1 Modelo de Dados

A abstração de dados é uma caraterística que permite a independência programa-dados e programa-operação. Neste contexto, o modelo de dados é um tipo de abstração de dados usados para prover uma representação conceitual. Conceito lógicos como objetos, entidades, suas propriedades e seus interelacionamento são usados em um modelo de dados, que podem ser mais fácies para os usuários entenderem os conceitos de armazenamento computacional [36]. Por esse motivo, o modelo de dados esconde detalhes de implementação e armazenamento, não interessantes para a maioria dos usuários de banco de dados.

De forma resumida, um modelo de dados é uma coleção de conceitos que podem ser usados para descrever um conjunto de dados e as operações para manipulá-los [37]. Os modelos de dados podem ser classificados, segundo a etapa de desenvolvimento do projeto de banco de dados em: conceitual, lógico e físico e estão intimamente relacionados com o ciclo de desenvolvimento de um projeto de banco de dados, onde a cada etapa, novas informações e detalhes são acrescidos. Nesse contexto, as informações pertencentes ao modelo são especificadas utilizando-se diferentes níveis de abstração, iniciados pelos de alto nível de abstração, como por exemplo, o Modelo de ER (*Entity Relationship*) até que sejam incorporados detalhes específicos, relacionados ao armazenamento dos mesmos..

Modelos de Dados Conceituais

O modelo conceitual de dados tem como característica básica à abstração da realidade, fornecendo uma base formal (de notação e semântica) com ferramentas e técnicas usadas para suportar a modelagem dos dados. Esse processo de abstração é onde somente os elementos essenciais da realidade observada são enfatizados, descartando-se os elementos não essenciais. Já o processo de modelagem conceitual de banco de dados compreende a descrição dos possíveis conteúdos dos dados, além de estruturas e de regras a eles aplicáveis. A seguir são apresentados os modelos de dados de Entidade Relacionamento, Entidade Relacionamento Estendido e o Orientado a Objetos, que são os principais modelos conceituais aplicados na área de bioinformática.

Modelo Entidade Relacionamento

O modelo ER foi apresentado por Peter Chen [38], a fim de representar as estruturas de dados de uma forma natural e mais próxima do mundo real. Este modelo tem sido usado amplamente para modelagem de dados. Os principais elementos do modelo ER, como próprio nome já diz, são as entidades, seus relacionamentos e seus atributos associados.

Uma entidade é um objeto que existe no mundo real e pode ser claramente identificada. As entidades podem ser classificadas em diferentes tipos, onde cada tipo contém um conjunto de propriedades comuns predefinidas. Há autores que preferem usar conjunto de entidades e entidade para designar um conjunto de objetos [36]. Neste contexto, um tipo de entidade contém um conjunto de entidades que satisfazem um conjunto de propriedades comuns predefinidas.

Por sua vez, um relacionamento é uma associação entre várias entidades. Formalmente, se E_1, E_2, E_3..., E_n são tipos de entidades, então um tipo de relacionamento R é um subconjunto do produto cartesiano E_1xE_2x...xE_n, então , $(e_1, e_2, ..., e_n) \mid e_i \epsilon E_1$, i=1,2,...,n

onde (e_1, e_2, ..., e_n) é um relacionamento. Na Figura 3.1 é apresentado o diagrama ER que descreve as entidades Gene, Exon, Segmento DNA e Intron, assim como também, os relacionamentos entre as mesmas. A entidade Gene contém segmentos de DNA, onde os segmentos de DNA podem ser Introns e/ou Exons que também são segmentos de DNA com um identificador unico, início e fim do segmento.

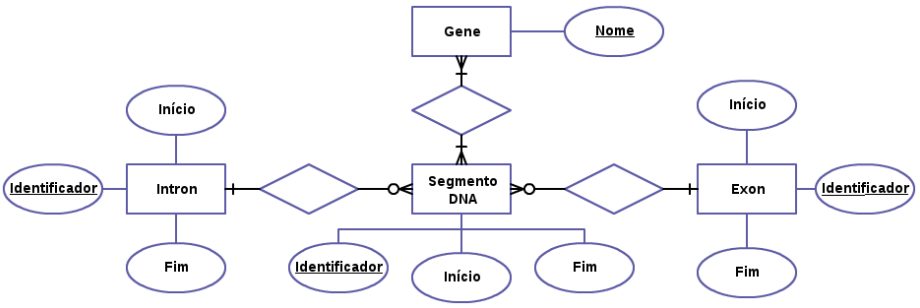

Figura 3.1: O Diagrama ER dos elementos que compõem o gene.

Os atributos descrevem as propriedades de cada entidade assim como as características que definem ou identificam a mesma dentro de um conjunto de entidades. Assim como as entidades possuem atributos, os relacionamentos também podem possuir.

Modelo Entidade Relacionamento Estendido

Ainda que os conceitos básicos do modelo ER modelem a maioria das características dos bancos de dados, alguns aspectos podem ser expressos de melhor forma por certas extensões do modelo ER básico [35]. Consequentemente, o modelo EER (*Enhanced Entity Relationship*) engloba todos os conceitos de modelagem ER (Entidade Relacionamento), além dos conceitos de subclasse e superclasse e especialização. A categoria ou tipo de união é um elemento importante do modelo EER, que é usado para representar uma coleção de objetos correspondentes à união de outros objetos de diferentes tipos de entidades e associados aos mecanismos de herança de atributos e relacionamentos.

Uma subclasse no modelo EER é um subgrupo de uma entidade, que é significativo e precisa ser representado explicitamente, em virtude de sua importância para as aplicações do banco de dados [36]. O conjunto que engloba esses subgrupos (subclasses) é chamado de **superclasse**. Um aspecto importante associado às subclasses é o da herança, pois, uma entidade, que é membro de uma subclasse, herda todos os atributos da entidade como membro da superclasse, assim como também herda todos os relacionamentos associados à superclasse.

A **especialização** define um conjunto de subclasses de uma entidade. Pode-se ter diversas especializações para a mesma entidade, baseada nas diferentes características que as distinguem. Em termos do diagrama ER, a especialização é representada por um componente triangular etiquetado com "*IS-A*". A relação "*IS-A*" pode-se chamar de relação superclasse/subclasse [35] já que, este tipo de relação começa desde superclasses genéricas até subclasses mais especificas, ou em outras palavras, entidade de alto nível a entidades de baixo nível (*top-down*). Contrariamente à especialização, a generalização acontece das entidades de baixo nível às entidades de alto nível, identificando características em comum

de um grupo de entidades e as generalizando em uma única superclasse, onde as entidades originais (baixo nível) são subclasses (*down-top*).

Outra limitação do modelo ER é o fato de não poder expressar uma relação de relacionamentos, por isso, a melhor forma de modelar uma situação dessas é usar o conceito de agregação. A agregação é a abstração para a construção de entidades compostas a partir de seus objetos componentes, esses objetos compostos podem ser tratados do mesmo jeito que qualquer entidade.

A Figura 3.2 apresenta um diagrama, com o modelo EER para os genes, segmentos de DNA, Intro e Exon e os seus relacionamentos. Como pode ser observado na figura, a entidade Intron e Exon estão representados como especialização de entidade segmento de DNA.

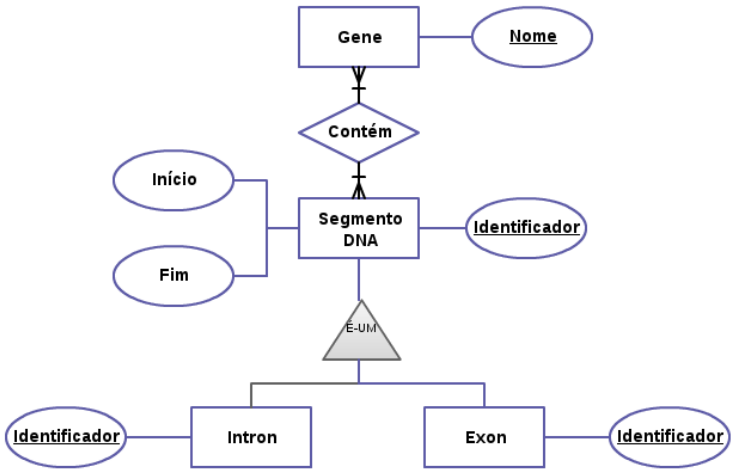

Figura 3.2: Diagrama EER do gene com os elementos que o compõem.

Modelos Orientado a Objetos

Assim como todo modelo de dados, o MOO (Modelo Orientado a Objetos) é uma abstração do mundo real. O MOO permite lidar com a complexidade inerente num problema do mundo real. O modelo orientado a objetos foi desenvolvido baseado no conceito de objetos, assim como o modelo ER foi desenvolvido baseado em entidades. O MOO é baseado no que é chamado "paradigma orientado objeto", onde tudo é modelado como objetos [39]. Para modelar sistemas complexos de maneira adequada tem-se o desenho orientado a funções e a abordagem orientada a dados (comumente usada por projetistas de banco de dados). Essas duas técnicas podem ser unidas em um só elemento chamadas classe, encapsulando tanto dados como processos [39]. Uma classe representa um conjunto de objetos parecidos, estes objetos têm propriedades (atributos) semelhantes e os mesmos comportamentos (operações), consequentemente a mesma semântica [40].

Define-se um objeto como um conceito, uma abstração, com limites nítidos e significado em relação à realidade estudada [40], por exemplo, a bactéria Escherichia Coli, o cromossomo 20 do genoma humano, o sequênciador illumina, dentre outros, são objetos dentro do mundo dos dados biológicos.

No MOO, um objeto pode ser qualquer coisa física ou abstrata que tem propriedades (atributos) intrínsecas ou comuns a diferentes objetos. Além disso, os objetos têm um conjunto de operações que definem seu estado. O estado de um objeto engloba suas propriedades (atributos e relacionamentos) e os valores que essas propriedades têm [41]. Já o comportamento de um objeto depende de seu estado e as operações que estão sendo desenvolvidas, estas operações são simplesmente ações.

Além dos conceitos de classe, objeto, propriedades e comportamento, tornam-se necessário outros conceitos chave que são discutidos a seguir:

- *Associação, ligação e multiplicidade*: a associação descreve um conjunto de ligações com estrutura e semântica comuns, a ligação é a conexão física ou conceitual entre instâncias de objetos, ou seja, é uma instância de uma associação. A multiplicidade especifica quantas instâncias de uma classe relacionam-se a uma única instância de uma classe associada, restringindo a quantidade de objetos relacionados. A multiplicidade pode ser expressa, de maneira geral, por "um" ou "muitos".
- Um *Atributo de ligação* é uma propriedade das ligações de uma associação.
- *Agregação*: A agregação é um tipo de associação forte onde um objeto agregado é constituído de componentes. A agregação é representada pelo relacionamento "parte-todo" ou "uma–parte-de" no qual os objetos que representam os componentes de alguma coisa são associados a um objeto que representa a estrutura inteira. Em termos semânticos, o objeto agregado é um objeto estendido tratado como uma unidade em muitas operações, embora fisicamente ele seja composto por objetos menores. Uma agregação é representada graficamente pelo símbolo de losango. A composição, por sua vez, é uma relação de agregação mais forte [42], onde a relação composição representa uma parte de um objeto que pertence a somente um objeto maior e existe e morre com o objeto maior. Por exemplo, um apartamento é parte de um somente um edifício.
- *Generalização, Especialização e Herança*: Generalização e especialização são dois diferentes pontos de vista do mesmo relacionamento, vistos a partir da superclasse ou das subclasses. Generalização deriva do fato de que a superclasse generaliza as subclasses. Especialização refere-se ao fato de que as subclasses refinam ou especializam a superclasse. A herança refere-se ao mecanismo de compartilhamento de atributos e operações utilizando o relacionamento de generalização.
- *Polimorfismo*: Trata-se da possibilidade de uma mesma operação atuar de modos diferentes em classes diferentes. Isto é possível quando uma operação for declarada em classes diferentes, porém com o mesmo nome, executando processamentos diferentes para atender os requisitos semânticos de sua classe.

A UML (*Unified Modeling Language*) é uma linguagem-padrão de modelagem e pode ser empregada para a visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas de *software* [43]. O desenvolvimento da UML foi baseado em técnicas de orientação a objetos, mas com influencias de outras técnicas. A UML é usada como uma metodologia de desenvolvimento, o que significa que ela não diz o que fazer e nem como projetar um sistema, mas ela auxilia a visualizar seu desenho e a comunicação entre objetos. Basicamente, a UML permite que desenvolvedores especifiquem, visualizem e construam os artefatos de sistemas [42].

Na Figura 3.3 é ilustrado um esquema orientado a objetos, onde o Intron e Exon são subclasses do segmento de DNA e a relação de composição entre as classes gene e segmento de DNA.

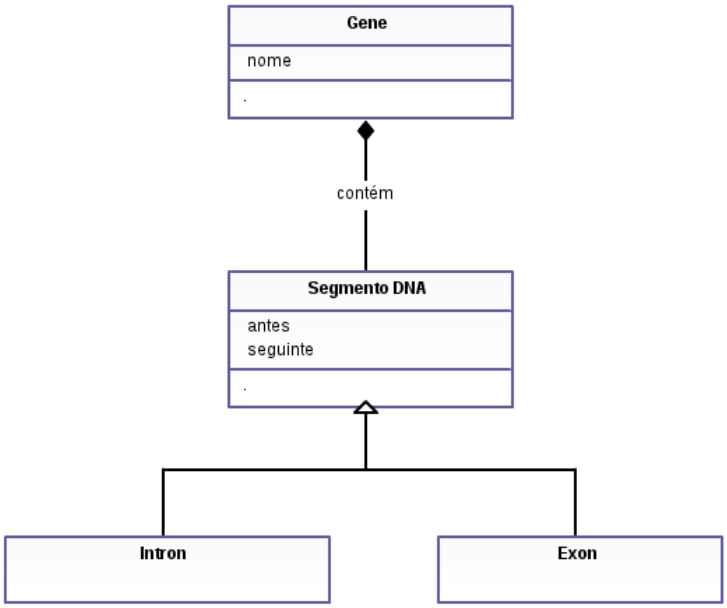

Figura 3.3: Diagrama da relação do gene com os elementos que o compõem usando o modelo orientado a objetos.

Modelo Relacional

O modelo relacional é um modelo de lógico muito usado atualmente. Esse modelo foi proposto por Edgar Codd [44] em 1970, como uma visão de apresentação dos dados. Codd mostrou que uma visão relacional dos dados permite a sua descrição natural, sem que sejam necessárias estruturas adicionais para sua representação, provendo uma maior independência dos dados em relação aos programas. Em complementação, apresentou bases para tratar problemas como redundância e consistência. Mais tarde, em outro trabalho [45], Codd definiu uma álgebra relacional e provou, por meio de sua equivalência com o cálculo relacional, que ela era completa, dando fundamentação teórica ao modelo relacional [45]. Este modelo, por suas características e por sua completude, mostrou ser uma excelente opção, superando os modelos mais usados àquela época: o de redes e o hierárquico. A maior vantagem do modelo relacional sobre seus antecessores é a representação simples dos dados e a facilidade com que consultas complexas podem ser expressas.

O modelo relacional tem por finalidade representar os dados como uma coleção de relações, onde cada relação é representada por uma tabela. Cada linha na tabela representa uma coleção de valores de dados, como uma tupla de uma relação [36]. Os valores de cada linha podem ser interpretados como fatos descrevendo uma instância de uma relação. Na terminologia do modelo relacional, cada tabela é chamada de relação; uma linha de uma

tabela é chamada de tupla; o nome de cada coluna é chamado de atributo; o tipo de dado que descreve cada coluna é chamado de domínio.

Um domínio D é um conjunto de valores atômicos (cada valor do domínio é indivisível). Durante a especificação do domínio é importante destacar o tipo e tamanho do atributo que está sendo especificado. Um esquema de relação R, denotado por R(A_1, A_2, ... , A_n), onde cada atributo A_i é o nome do papel desempenhado por um domínio D no esquema relação R, onde D é chamado domínio de A_i e é denotado por dom(A_i). O grau de uma relação R é o número de atributos presentes em seu esquema de relação.

A instância r de um esquema de uma relação denotado por r(R) é um conjunto de n-tuplas r = [t_1, t_2, ... , t_n] onde os valores de [t_1, t_2, ... , t_n] devem estar contidos no domínio D. O valor nulo também pode fazer parte do domínio de um atributo e representa um valor não conhecido para uma determinada tupla.

Dois conceitos fundamentais de um modelo relacional são chave primária e chave estrangeira. Chave primária é utilizada para identificar unicamente uma tupla em uma realização. Chave estrangeira é utilizada para identificar os relacionamentos entre as tabelas. Neste contexto, a restrição de domínio especifica que, dentro de cada tupla, o valor de cada atributo A deve ser um valor atômico do domínio Dom(A). A restrição de chave define que toda tupla tem um conjunto de atributos que a identifica de maneira única na relação, isto é, nenhum valor de chave primária poderá ser repetido. A restrição de chave estrangeira define que uma relação pode ter um conjunto de atributos que contém valores com mesmo domínio de um conjunto de atributos que forma a chave primaria de outra relação. Este conjunto é chamado de chave estrangeira. Na Figura 3.4 é apresentado um diagrama relacional do Gene, o o classe Gene esta composto por Segmento de DNA (através de una relação de composição). A classe Segmento de DNA tem una relação de especialização com as classes Intron e Exon.

Na Figura 3.4 é apresentado um diagrama no modelo relacional do Gene, Segmento de DNA, Intron, Exon e seus relacionamentos.

3.1.2 Modelos de Dados para Bioinformática

Para o gerenciamento de dados biológico é necessário um claro entendimento da natureza dos dados. Perguntas tais como: "Que tipo de dados serão armazenados? Que tipo de relacionamentos tem-se entre esses tipos?" devem ser respondidas antes da implementação real. A modelagem de dados conceituais pode prover uma forma científica para capturar as principais propriedades dos dados biológicos. Os modelos de dados estudados anteriormente têm uso extensivo para a modelagem de dados biológicos. Nessa seção, são apresentados diferentes trabalhos que usam modelos de dados tais como o modelo entidade relacionamento, modelo entidade relacionamento extendido e o modelo orientado a objetos nas aplicações de gerenciamento de dados de projetos na área de bioinformática.

Em estudos preliminares, foram identificados trabalhos que apresentam propostas relacionadas aos objetivos desta dissertação. A maioria deles surge com o intuito de procurar uma forma de representar conceitos da biologia molecular.

Nos próximos parágrafos são detalhados alguns dos principais modelos de dados para representar dados biológicos disponíveis na literatura, suas características, vantagens e eventuais desvantagens. Pretende-se, dessa forma, justificar o modelo proposto nesta dissertação como alternativa relevante aos modelos existentes.

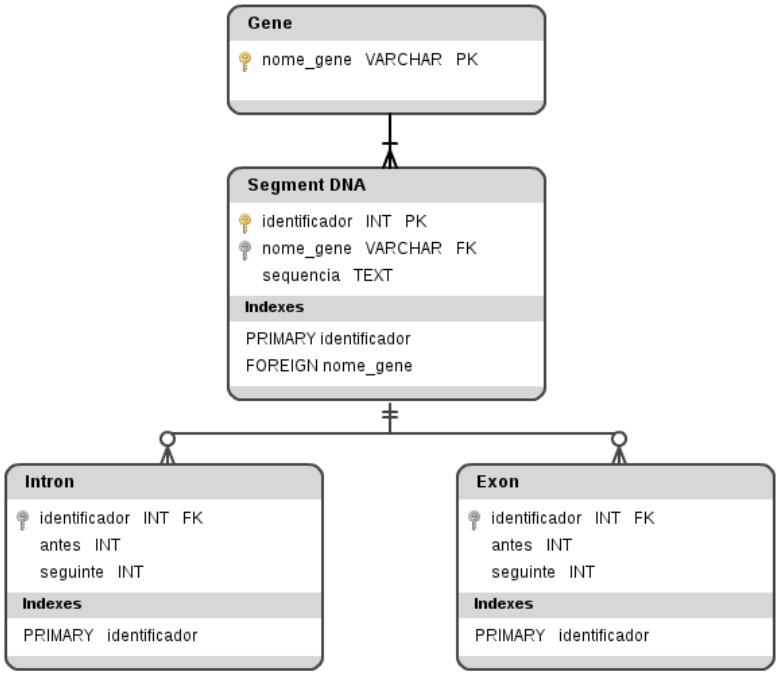

Figura 3.4: Diagrama da relação do gene com os elementos que o compõem usando o modelo relacional.

1. Paton *et al.* (2000) [5] é um dos primeiros trabalhos que apresentou modelos conceituais que descrevem dados genômicos e transcritômicos de eucariotos. Os modelos conceituais deste trabalho são descritos usando diagramas de classes usando UML. Neste trabalho são apresentados uma coleção de modelos conceituais para dados de sequências genômicas. Além disso, são representados conceitos relacionados aos acontecimentos naturais ou modificações induzidas ao genoma, descrevendo a modificação e as consequências dessas modificações. Dessa forma, permitindo a integração qualitativa e quantitativa dos distintos conjuntos de dados genômico funcionais que tem sido produzidos. A representação de sequências genômicas é feita por meio de um esquema básico onde a entidade genoma é composta pela entidade cromossomo, a entidade cromossomo é composta pela entidade fragmentos de cromossomo que ao mesmo tempo está composta por regiões transcritas e não transcritas e o nível de granularidade vai aumentando. Este modelo em especifico é importante pois representa detalhes das sequências de DNA (genômica) e RNA (transcritômica) até serem traduzidas em proteínas.

2. Bornberg-Bauer e Paton (2002) [46] fazem uso de conceitos básicos dos modelos ER e modelos orientado a objetos para especifizar modelos conceituais no contexto da bioinformática. Pode-se considerar uma extensão dos modelos apresentados em [5], pois, além de apresentar o modelo geral para sequências genômicas (Figura 3.5), são apresentados modelos para estruturas de proteínas e motifs usando os modelos ER e MOO. O modelo ER é usado para representar a relação que existe entre enzima, proteína e DNA com biopolímeros, assim como a relação de enzima-proteínas e enzima-reação. Embora seja usado o modelo ER, é apresentado o mapeamento desse

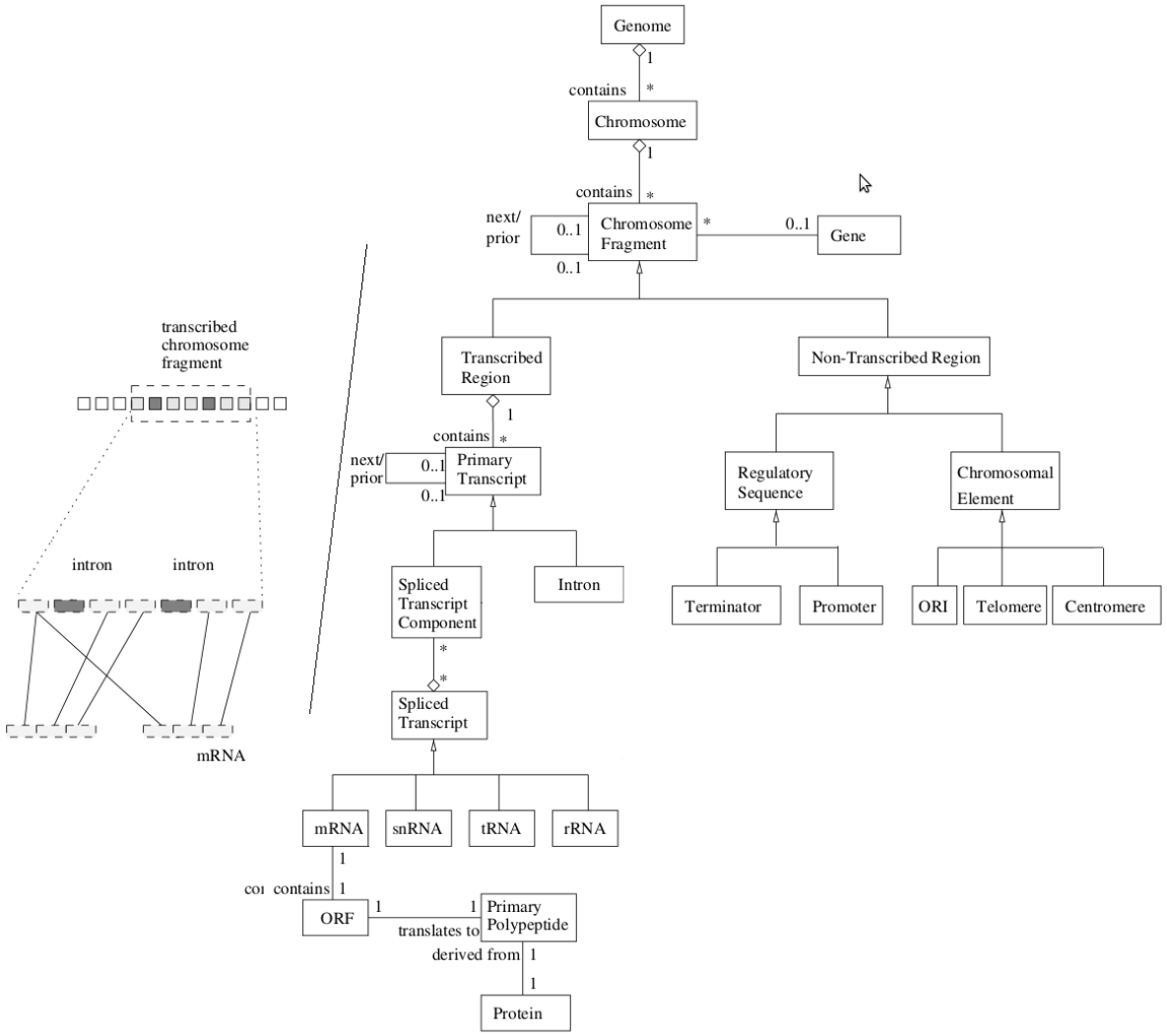

Figura 3.5: Diagrama para dados genômicos [5].

modelo ao seu equivalente no modelo relacional. Para representar a estrutura da proteína é usado o diagrama de classes (UML), detalhando elementos da estrutura secundária e terciária da proteína.

3. Elmasri *et al.* (2006) [6] apresenta modificações no modelo EER para representar de melhor forma algumas características especiais da biologia tais como, sequências ordenadas, processos de *input/output*, e características espaciais das moléculas. Dados de sequência como os ácidos nucleicos DNA/RNA e amino ácidos das proteínas, ambos tem esta propriedade de ordem. Processos importantes como expressão de genes, metabolismo, transcrição e tradução, envolvem muitas entidades biológicas, eventos ordenados e processos de *input/output*. Para acomodar estas características esta abordagem fez algumas modificações no modelo EER introduzindo três tipos especiais de relacionamentos: relação de ordenação, relação de processo e relacionamento de molécula espacial.

A *relação de ordenação* representa a ordem dos elementos das sequências de DNA e proteínas, onde a característica de ordem é muito importante pelo fato que mudanças na ordem tem grande impacto em níveis superiores da estrutura e na sua função. A *relação de processo* representa o comportamento dinâmico dos diferentes agentes. Por exemplo, o mRNA é a saída do processo de transcrição e a entrada do processo de tradução. A *relação espacial* é definida para descrever relações entre um conjunto de átomos no espaço 3D, onde a função é determinada pela sua estrutura tridimensional, por exemplo a estrutura do DNA afeta as regiões que podem ser lidas para criar proteínas. A Figura 3.6 mostra a notação destas 3 novas relações.

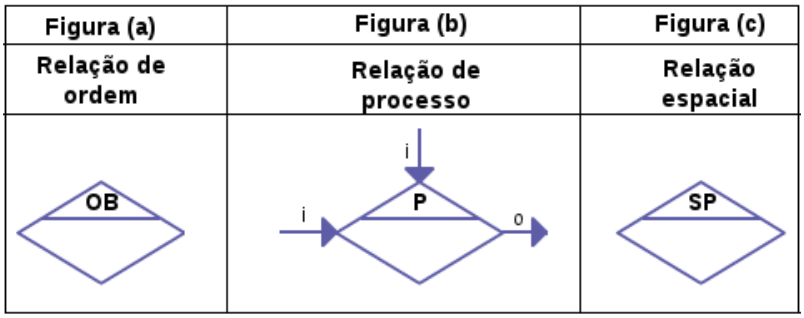

Figura 3.6: Notação para as relações de ordem, processo e espacial [6]

4. Busch e Wedemann (2009) [7] definiram um DOM (*Dynamic Object Model*) [47] baseado no modelo orientado a objetos. Neste trabalho uma coleção de modelos foram especificados no intuito de ter um modelo flexível suficiente para o domínio da biologia molecular, pretendendo suportar tanto a mutabilidade como também a interoperabilidade entre diferentes tipos de dados. Este modelo é composto de quatro modelos: operacional, conhecimento, meta modelo e o de informação. Cada um desses modelos são descritos a seguir.

   O *modelo operacional* define o alcance do domínio do modelo. Contém conceitos abstratos fundamentais da biologia molecular. Este modelo é um modelo de classes orientado a objetos de diferentes tipos de moléculas como por exmplo, DNA, RNA, proteína.

   O *modelo de conhecimento* define conceitos concretos da biologia molecular; estes conceitos e suas relações sujeitos à modificação do conhecimento da biologia molecular.

   O *meta modelo* define a estrutura do modelo de conhecimento. Une conceitos concretos definidos no modelo de conhecimento com conceitos abstratos do modelo operacional.

   O *modelo de informação* contém dados da aplicação que são originados durante a execução. Contém instâncias das classes do *modelo operacional*.

   O *modelo operacional* e o *metamodelo* definem a estrutura dos modelos de informação e conhecimento respectivamente (veja Figura 3.7), observando que o modelo de conhecimento contém os dados concretos do modelo operacional onde podem ser

realizados modificações. O modelo de conhecimento é a chave da flexibilidade da abordagem.

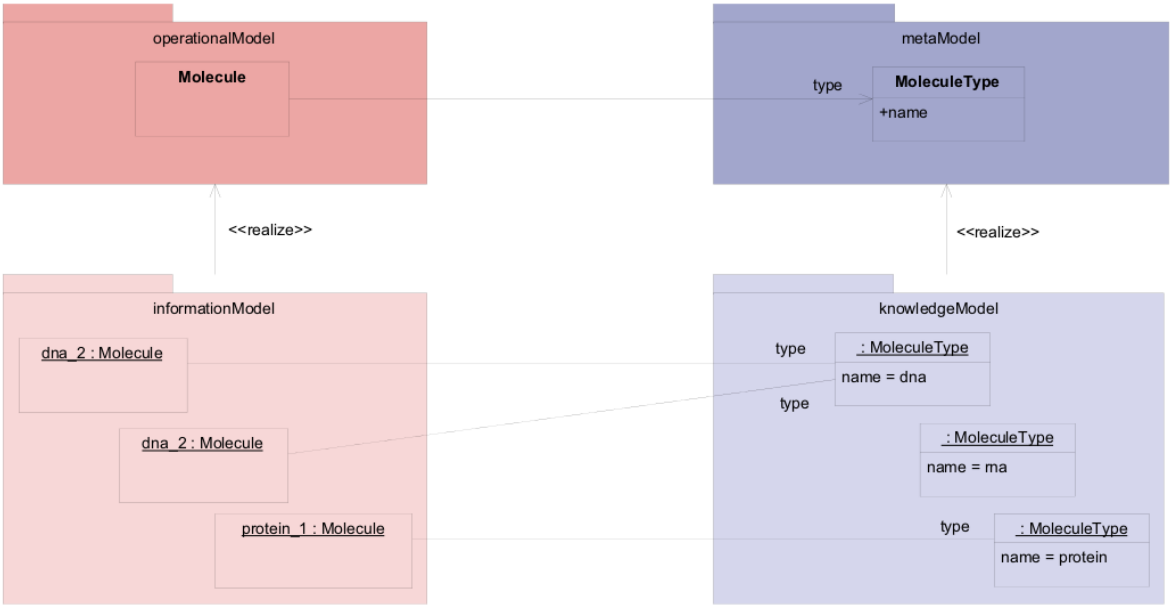

Figura 3.7: Os quatro submodelos: modelo operacional, meta modelo, modelo de conhecimento e modelo de informação [7].

5. Macedo (2007) *et al.* [8] propõe uma linguagem conceitual chamada BioConceptual. A *BioConceptual* propõe estender os construtores tradicionais (conceito, relação e classificação) do MOO para dessa forma melhorar sua expressividade e facilitar a especificação do domínio biológico em termo de dados. *BioConceptual* proporciona uma notação gráfica associada para cada tipo de construtor. Neste contexto, algumas extensões são:

   Tipo de dado objeto: é um tipo de abstração que permite representar o domínio da aplicação em termos de dados, similar a qualquer linguagem tradicional. Por exemplo, no esquema pode-se ter o conceito de Exon, ele usará o construtor de tipo de dados objeto para definir este conceito.

   Atributo de um tipo de dado objeto: permite a definição de tipos de atributos simples (comumente usados) e complexos que denota um conjunto de atributos que podem ser tanto complexos como simples.

   Tipo de relacionamento: o relacionamento determina uma ligação entre dois ou mais tipos de objetos, disponibilizando os relacionamentos "é-um", "parte-todo" e associação. A partir desses relacionamentos pode-se usar o construtor *Constraint* para aumentar a semântica dos outros contrutores do *BioConceptual*.

   Relacionamentos de associação: associações no *BioConceptual* são ligações direcionadas dada a necessidade de indicar qual é a ordem dos parâmetros dentro do predicado que representa o relacionamento. As instâncias dos relacionamentos não

podem ter papéis pendentes. Por outro lado, tem-se outras restrições, como as cardinalidades que caracterizam cada um dos papéis envolvidos.

Ligações "É-UM" entre tipos de objetos: referenciado como relacionamento de generalização/especialização.

Relação de agregação: define um construtor especial chamado *configuration constraint*, que ajuda a especificar uma configuração usando relacionamentos de agregação, onde pode ser usado uma expressão regular.

Restrições de integridade: define-se *Constraint* como construtor específico de restrições, podendo ser definida usando a lógica de primeira ordem. O construtor *Constraint* pode ser aplicado a todos os outros tipos de construtores.

Múltiplas percepções e representações: *BioConceptual* propõe um construtor chamado *perception* que objetiva a especificação das percepções (diferentes formas de ver o mesmo fenômeno) dos cientistas. A ideia geral é associar tipos de objetos, atributos ou tipos de relacionamentos com percepções.

A Figura 3.8 mostra o uso do construtor *configuration constraint* para definir a configuração de uma região de transcrição que é composta por regiões intercaladas de introns e exons. É mostrado que uma região de transcrição é uma sequência que começa com um exon e pode ter um o mais exon seguidos por um e só um intron.

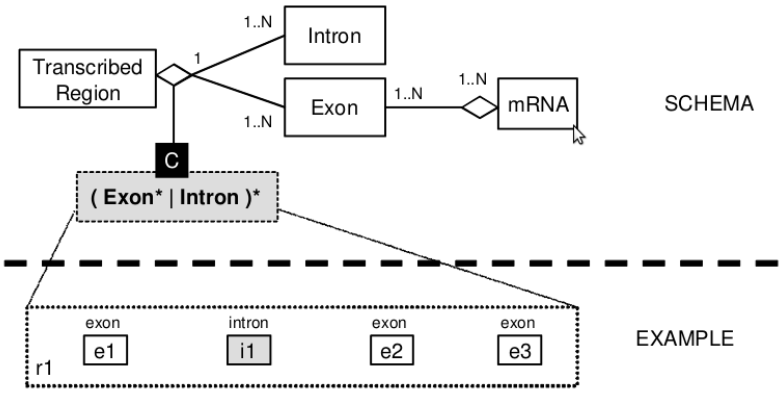

Figura 3.8: Definição de uma ordem entre instancias de tipo agregação [8].

Comparação Entre os Modelos

Como apresentado nesse capítulo existem diferentes propostas para modelar conceitualmente dados biológicos. De maneira resumida a Tabela 3.1 faz uma comparação entre os modelos de dados utilizados, assim como também, a dificuldade do seu uso. Em relação a dificuldade de uso foi utilizado o seguinte critério: baixo, quando não houve variação das propostas de modelagem tradicional; médio quando foram apresentadas extensões; alto, quando além das extensões novas técnicas devem ser aprendidas para a conclusão do modelo.

Os modelos conceituais representam conceitos da biologia molecular. Os modelos [5, 46, 7] usam os modelos de dados existentes (ER, EER e MOO). Outros modelos tais

Tabela 3.1: Comparação dos modelos conceituais. A modelo de dados que usa, dificuldade no uso, plataforma onde foi implementada.

		Abordagem	**Dificuldade de uso**	**Implementação**
1	Paton *et al.* [5]	MOO	Baixa	PEOT
2	Bornberg-Bauer e Paton [46]	ER-MOO	Baixa	PEOT
3	Elmasri *et al.* [6]	ER-EER	Medio	Qualquer SGBD relacional
4	Busch e Wedemann [7]	MOO-DOM	Alto	MCK
5	BioConceptual [8]	MOO	Medio	*Framework* orientado a objetos

como [6, 8] adicionam novas características aos modelos de dados para adaptar-se e representar conceitos complexos da biologia molecular. O fato de modificar modelos de dados e acrescentar algumas propriedades tem o objetivo de representar conceitos difíceis de modelar com os modelos de dados existentes. As modificações tem o objetivo de simplificar a representação de conceitos complexos. No etanto a implementação das abordagens foram usados o banco de dados orientado a objetos POET (agora FastObjects de Versant que comprou *Poet Software*), banco de dados relacionais, o *framework* MCK (*Molecular Computer Kit*) [7] e *framework* orientado a objetos para as respectivas abordagens.

3.1.3 Proposta de Esquema de Dados para Bioinformática

Assim como os modelos de dados já apresentados, tem-se esquemas de banco de dados para gerenciar dados biológicos. Enquanto os modelos de dados estão mais interessados em representar os dados biológicos sem preocupar-se na implementação, os esquemas relacionais são desenvolvidos baseados num modelo de implementação que neste caso seria o modelo relacional. Os esquemas relacionais tomam em consideração requerimentos como o SGBD onde será implementado. Entre os esquemas de banco de dados relacionais usados pelos projetos da bioinformática destacam-se: o GUS [9] e o CHADO [48].

GUS

O GUS(*Genomics Unified Schema*) é um esquema de banco de dados relacional que suporta uma ampla gama de tipos de dados que inclui genômicos, expressão de genes, regiões de transcrição, proteômica, entre outros [9]. O GUS propôs uma modelagem de dados para a implementação de aplicações bioinformáticas, de modo que o núcleo central do modelo é baseado no dogma da biologia molecular (ver Seçao 2.1.3). Conforme a Figura 3.9 apresenta, as entidades principais e suas relações são: um gene pode ter vários RNAs, um RNA pode dar origem a várias proteínas. O GUS também separa as anotações dos genes das anotações de RNAs.

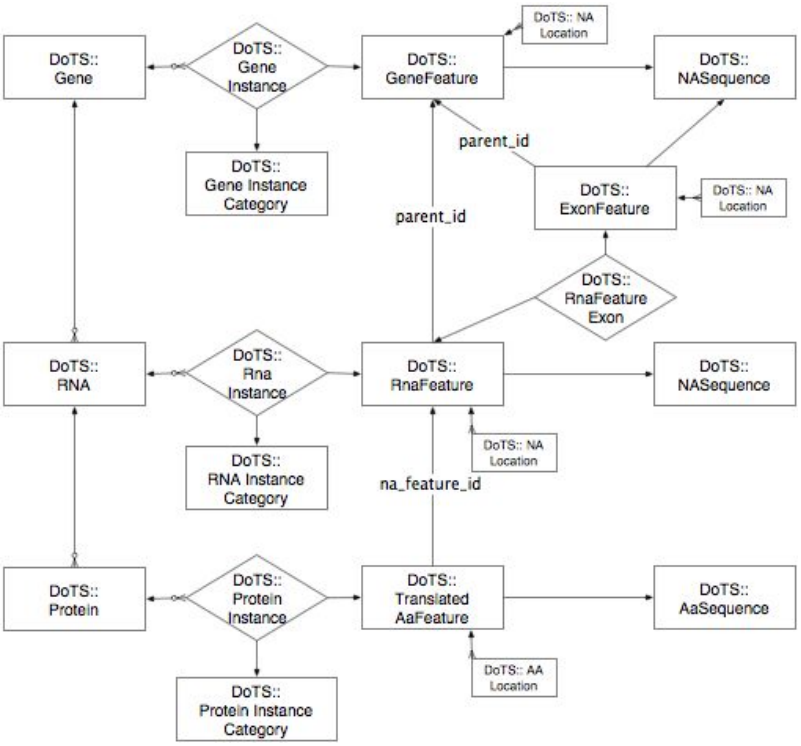

Figura 3.9: Diagrama ER representando o dogma central da Biologia Molecular [9].

O esquema relacional GUS é implementado através de sete esquemas relacionais: DoTS (*Database of Transcript Sequence*); RAD (*RNA Abundance Database*); TESS (*Transcription Element Search System*); SRes (*Source Shared*); e o núcleo; usado para rastreamento não biológico e sobrecarga. Mas o número de tabelas que o esquema GUS possui é aproximadamente de 480 tabelas [9], o que limita seu entendimento e consequentemente a realização de consultas nesse esquema é muito complexa [49]. De forma similar a outro esquemas, os usuários do GUS devem avaliar o esquema que se acomoda de melhor forma a suas necessidades [49].

CHADO

O CHADO é um esquema de banco de dados relacional modular usado para administrar dados biológicos para uma grande variedade de organismos, especialmente, informação que está diretamente ou indiretamente envolvida com sequências DNA, sequências de RNA e proteínas [10, 50, 48]. O CHADO é baseado na metodologia orientado a ontologias e terminologias a qual é a chave da sua flexibilidade.

O CHADO foi originalmente desenvolvido para integrar recursos de informação em dois bancos de dados de Drosophila independentes. Desde então, tem sido desenvolvido um esquema de banco de dados genômico ontológico em resposta ao *feedback* dos usuários finais e da comunidade de bioinformática. É parte integrante como um componente importante no projeto GMOD (Modelo de Banco de Dados Genérico para Organismos), e agora fornece a infraestrutura de banco de dados para numerosos pacotes de software dentro e fora do projeto GMOD (*The Generic Model Organism Database*) [50].

A modularidade é um princípio fundamental que reduz a complexidade e as dependências. Neste contexto, o CHADO tem cinco módulos centrais: de uso geral, publicação, auditoria, vocabulário controlado (ontologias) e de sequência. O *módulo de uso geral* prove entidades de dados com identificadores estáveis, globais e únicos. A tabela *dbxref* armazena os identificadores, junto com uma coluna que referência o nome do banco de dados, que é armazenado em uma tabela separada. O *módulo de publicação* é definido para armazenar informações de proveniência de dados. Neste módulo, a tabela *pub* não esta limitado a armazenar informação de documentos publicados, mas também comunicações pessoais e análises. O *módulo de auditoria*, é autogerado pelo esquema de banco dados mesmo. Para cada tabela do banco de dados existe um conjunto de triggers que populam a tabela *audit_ chado*. Uma vez realizado uma insertação, atualização, ou deleção é armazenada dentro da tabela de auditoria o tempo, e o identificador de usuário. O *módulo de ontologias e vocabulário controlado* são parte integrante do CHADO que permite ter um esquema genérico que tipifica todas as entidades dentro do banco de dados. A tabela *cvterm* armazena cada um desses tipos (dados e relações). O módulo sequência, mais particularmente a tabela *feature* é muito importante para que o esquema do CHADO gerencie sequências de dados. Neste contexto uma *feature* é uma região de uma macromolécula (DNA, RNA ou proteína) [10, 50]. A Figura 3.10 mostra as tabelas mais importantes que compõem a módulo de *sequence feature*.

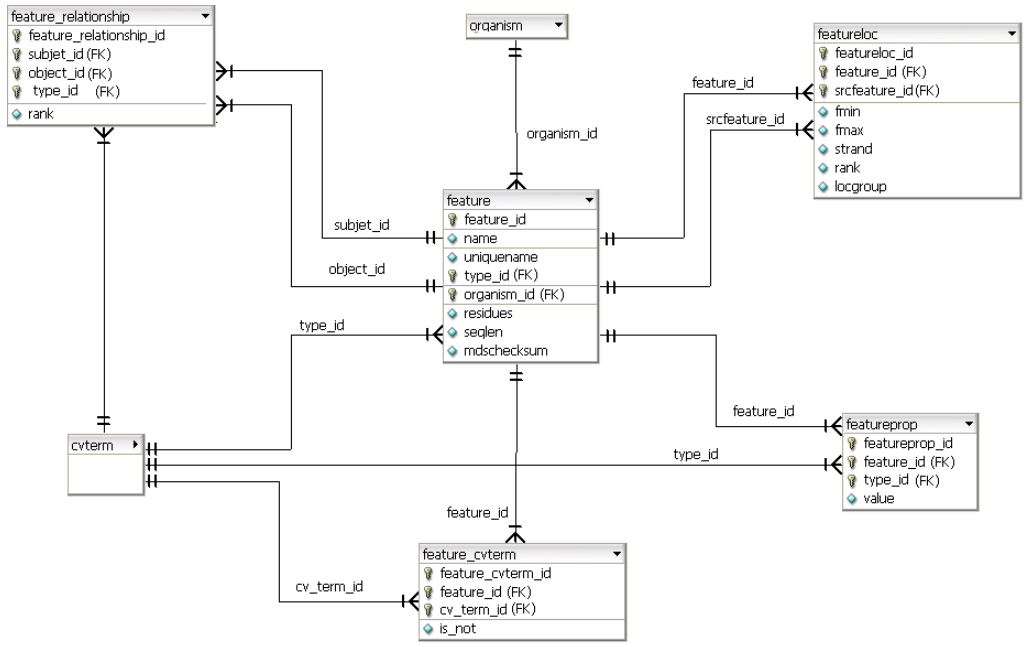

Figura 3.10: Esquema mostra as principais tabelas do módulo de sequência. Algumas tabelas e colunas foram omitidas para fazer o diagrama mais conciso. Adaptado de [10].

Capítulo 4

Modelo de Dados para um *Pipeline* de Sequenciamento de Alto Desempenho

O presente capítulo apresenta um modelo de dados orientado a objetos para dar suporte a um *pipeline* de sequenciamento de alto desempenho usando a notação UML. Além disso, é apresentado o esquema relacional correspondente. O objetivo desta proposta é oferecer um modelo capaz de representar as diferentes fases que envolvem um projeto de sequenciamento transcritômico. Dessa forma, tentar trazer o modelo conceitual mais perto do domínio do processo de sequenciamento, além do domínio biológico.

Como exposto no capítulo anterior (Seção 3.2), os modelos conceituais disponíveis na literatura, tem o foco principal no dado biológico, e não, no processamento dos projetos de sequenciamento atuais. O objetivo desta dissertação é propor o modelo de dados conceitual que possa integrar a modelagem de conceitos próprios da biologia molecular assim como a modelagem dos processos envolvidos no sequenciamento de alto desempenho transcritômico. O primeiro passo para atingir este objetivo é a definição da estrutura do *pipeline*, descrita na Seção 4.1. O próximo passo é a definição dos modelos conceituais para cada uma dessas fases, assim como o modelo conceitual geral para o *pipeline* de sequenciamento de alto desempenho transcritômico detalhado na Seção 4.2. Na Seção 4.3 é desenvolvido o esquema relacional para a implementação do modelo de dados proposto em um sistema gerenciador de banco de dados relacional.

4.1 Estrutura Geral do *Pipeline* de Sequenciamento de Alto Desempenho

O objetivo do modelo conceitual apresentado nesse trabalho é dar suporte aos projetos de sequenciamento de alto desempenho transcritômico. Os novos sequenciadores de alto desempenho produzem SRS de comprimento que variam de 30 pb a 400 pb [16]. Em contraste as sequências de comprimento maior do sequenciamento Sanger, o pequeno tamanho das SRS produzido pelos novos sequenciadores torná-lo mais difícil para realizar as diferentes análises de um *pipeline* tradicional. Além disso, para a montagem de sequências e o resequenciamento de genoma, sequências mais curtas vão exigir uma maior cobertura ou amostragem do genoma para representar com precisão as informações genéticas [16].

É proposto um *pipeline* de três fases (Figura 4.1): filtragem, mapeamento e análise. A estrutura do *pipeline* permite voltar a uma fase anterior de acordo com a necessidade dos usuários do projeto. Por exemplo, pode-se voltar da fase de mapeamento à fase de filtragem, da fase de análise à fase de mapeamento ou filtragem.

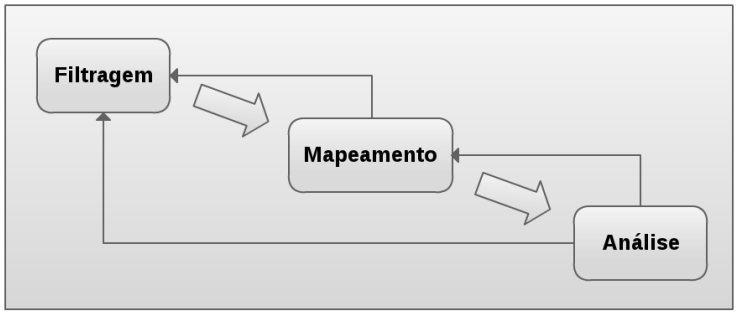

Figura 4.1: Estrutura do *pipeline* de alto desempenho com as fases da filtragem, mapeamento e análise

Fase de filtragem: Após o processo de sequenciamento feito pelo sequenciador de alto desempenho, milhões de fragmentos de DNA/RNA são gerados. O processo de filtragem é fundamental, pois geralmente há sequências que apresentam DNA/RNA de regiões cujo sequenciamento é impreciso, ou simplesmente tem regiões que não são interessantes seu processamento nas fases subsequentes. O processo de sequenciamento gera as SRS junto com as sequências de qualidade associadas a cada base. A qualidade é um valor numérico que expressa a probabilidade de erro associada a cada base. Cada projeto fixa um valor mínimo aceitável de qualidade. As bases que apresentam um valor de qualidade abaixo de um limite preestabelecido devem ser descartadas, uma vez que podem gerar imprecisões nas etapas subsequentes do *pipeline*. As SRS geradas também podem conter contaminantes provenientes de fragmentos de DNA não pertencentes à espécie estudada.

Em um laboratório é comum a execução de experimentos com organismos diferentes. Acidentalmente, é possível que uma amostras seja contaminada com sequências de outro organismo estudado no mesmo laboratório. Outra possibilidade de contaminação ocorre quando se estuda organismos que vivem relações simbiontes ou é atacado por alguma doença. Existe a possibilidade de contaminação, pois durante a coleta de material existe a possibilidade da obtenção de DNA de ambos [25].

No sequenciamento são usados outros fragmentos tais como *primers*, vetores e adaptadores que podem de alguma forma conter contaminantes e afetar o valor de certeza das bases sequenciadas (qualidade) [24]. Outras características possivelmente presentes nas sequências transcritas que podem dificultar o processamento das etapas subsequentes do *pipeline* são as presenças de caudas poli-A/poli-T (longa sequência de nucleotídeos adeína e timina) e repetições de elementos [25]. Nesta fase de filtragem, os contaminantes, regiões de baixa qualidade, caudas poli-A e poli-T e repetições de elementos são removidos das sequências. Se depois desse processo o tamanho das sequências ficar abaixo de um limite pré-estabelecido, as mesmas são descartadas, não sendo utilizadas nas etapas subsequentes do *pipeline*.

Os parâmetros utilizados para a filtragem variam, de acordo com a espécie estudada, com os objetivos do projeto e com a experiência dos pesquisadores. O limite mínimo para

que uma base seja considerada de baixa qualidade e o tamanho mínimo para que uma sequência não seja descartada após a limpeza são exemplos de parâmetros configuráveis nessa etapa.

Fase de mapeamento: Uma vez que as sequências obtidas pelos novos sequenciadores são relativamente curtas em relação ao sequenciamento tradicional, isso torna inviável o uso das técnicas tradicionais para reagrupar e ordenar os fragmentos sequenciados no DNA original, de forma a corresponderem às suas respectivas posições nos cromossomos [4]. Nesta etapa, usa-se um genoma de referência, normalmente um organismo próximo ao organismo sendo sequenciado cujo genoma já é conhecido com grande precisão. Dado esse genoma de referência, pode-se mapear as pequenas sequências obtidas pelos novos sequenciadores e agrupá-las conforme a posição das mesmas no mapeamento. Uma vez que as sequências agrupadas constituem um número muito menor a ser analisado e visto possuírem poucas diferenças entre si, pois estão mapeadas aproximadamente na mesma região do genoma, seria possível aplicar técnicas de montagem tradicional a esses grupos de sequências. Além de um genoma de referência, seria possível também utilizar bibliotecas de exons como sequências de referências para a verificação de *splicing* alternativo a partir do sequenciamento das SRS [1].

A tarefa de mapeamento de SRS é buscar a localização onde uma SRS é idêntica à referência. Porém, na verdade a referência nunca é uma representação perfeita da fonte biológica atual do DNA/RNA que foi sequenciado. Além disso, as SRS podem as vezes ser mapeadas perfeitamente em vários locais [51]. Portanto, a verdadeira tarefa dessa fase é encontrar o local onde cada SRS seja mapeada com mais alta precisão no genoma de referência. Comumente, esta fase é incluída no *pipeline* transcritômico quando existem estudos filogenéticos de organismos próximos que tenham sido bem estudados, chamados genomas de referência. As SRS que são mapeadas na mesma região do genoma de referência são agrupadas dentro de um conjunto que é representado por uma sequência de consenso construída a partir do todas as SRS que pertencem a este conjunto.

Fase de análise: Esta fase tem uma grande dependência do propósito do projeto. A fase de análise é o processo de procurar informação relevante das SRS obtidas na fase do mapeamento, devidamente interpretadas, para extrair seu significado biológico e colocá-lo no contexto da compreensão dos processos biológicos [52]. Esta fase é útil para a formulação de testes de hipóteses biológicas [53]. O processo de análise, normalmente contém um passo de anotação onde funções (bioquímicas e biológicas) são atribuídas a um grupo de sequência. Um resultado da procura é encontrar informação relevante, como sequências de genes e regiões reguladoras, identificação de expressão de genes, análises de filogenia, assim como outras análises.

4.2 Modelo Conceitual para o *Pipeline* de Sequenciamento de Alto Desempenho

Seguindo as fases descritas por Silberschatz et al. [35], a fase inicial do projeto prevê entrevistas com especialistas do conhecimento, neste caso especialistas em biologia molecular para definir e caraterizar o problema. Neste trabalho, o objetivo é desenvolver modelos de dados para um *pipeline* de sequenciamento de alto desempenho para armazenar os dados gerados pelas diferentes fases do *pipeline*.

A segunda fase prevista por Silberschatz et al. [35] se refere à escolha do paradigma de modelamento para a modelagem conceitual; neste caso, escolheu-se o modelo MOO por sua capacidade de representar dados complexo, e a seguir foi desenvolvido o diagrama de classes usando usando a notação UML (veja Figura 4.2). O modelo de dados conceitual está dividido em três modelos: filtragem, mapeamento e análise. Os modelos estão de acordo com as fases do *pipeline* apresentado anteriormente. Cada modelo é apresentado nas seções seguintes.

A Tabela 4.1 apresenta o nome de todas as entidades do modelo, assim como também, a sus descrição.

Tabela 4.1: Entidades de cada modelo

Nome entidade	**Descrição entidade**
organism	Organismos a serem estudados
sample	Amostras tiradas de algum organismo
project	projetos
short_read	Sequências de bases
sequencer	Sequenciadores
quality_type	Tipo de qualidade usado pelo sequenciador
filtering_process	Processo de filtragem das sequências
parameter	Parâmetros usados no processo
filtering_parameter	Valores dos parâmetros
reference_genome	Genomas de referência bem anotados
chromosome	Cromossomos do genoma de referência
gene	Genes contidos nos fragmentos de cromossomos
chromosome_fragment	Segmentos de cromossomo
mapping_process	Processos de mapeamento
mapping_result	Fragmentos (sequências) mapeados
database	Bancos de dados usados
analysis_process	Processos de análises
ncRNA_identification	Identificação de RNA não codificadores
differential_expression	Análises de expressão diferencial
phylogenetic_analysis	análises filogenéticas (homologia, hortologia e paralogia)
sequence_alignment	Alinhamento de sequências
other	Outras possíveis análises

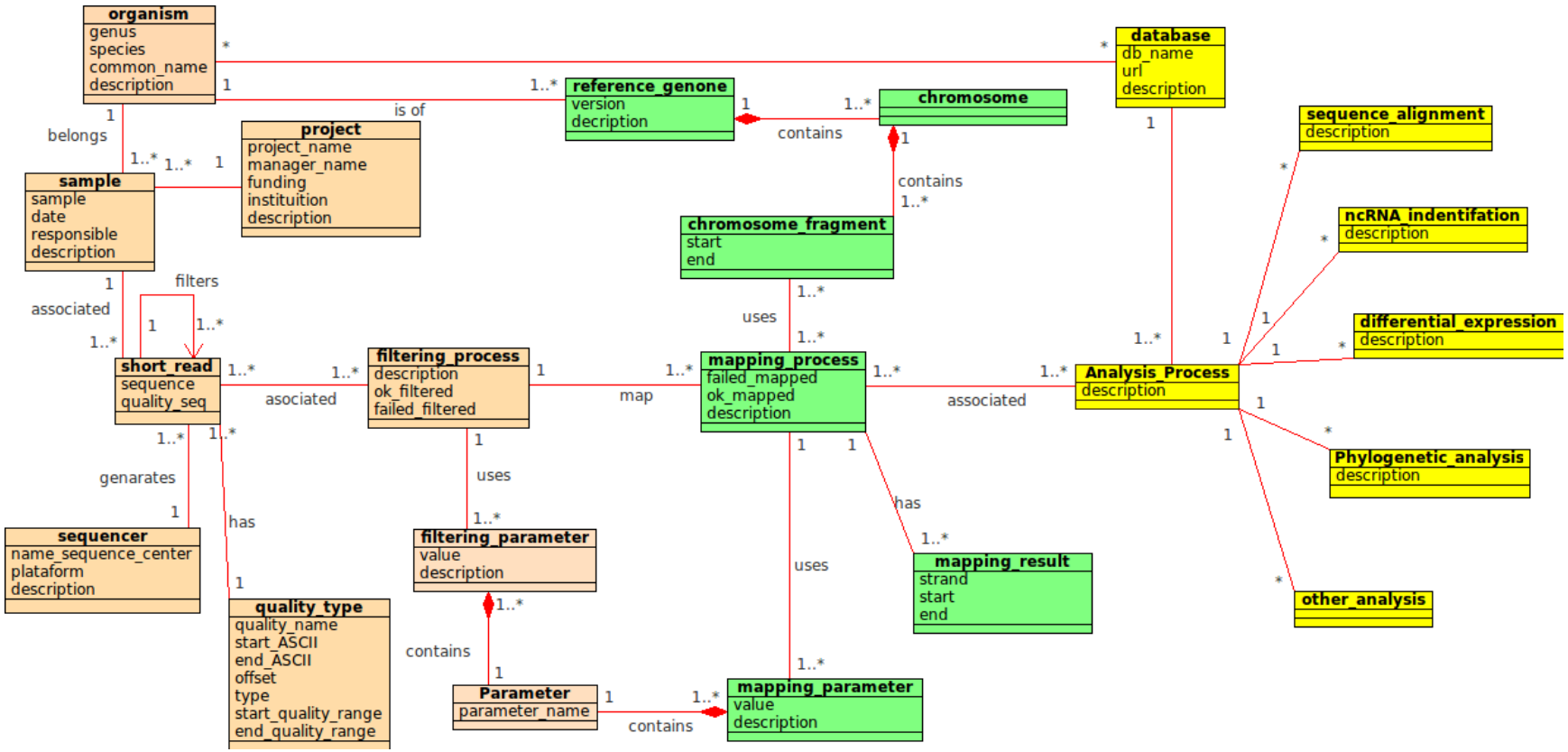

Figura 4.2: Diagrama de classes do modelo conceitual para um *pipeline* de sequenciamento de alto desempenho transcritômico. Ver diagrama ampliado no anexo I.

O modelo de dados está dividido em três modelos menores. Estes modelos representam as fases da filtragem, mapeamento e análise. A Tabela 4.2 mostra para cada modelo as entidades que o compõem.

Tabela 4.2: Entidades do modelo do *pipeline*

Nome modelo	**Nome entidade do modelo**
Filtragem	*organism*
	sample
	project
	short_read
	sequencer
	quality_type
	filtering_process
	filtering_parameter
	parameter
Mapeamento	*reference_genome*
	chromosome
	chromosome_fragment
	mapping_process
	mapping_parameter
	mapping_result
Análise	*database*
	analyse_process
	ncRNA_identification
	differential_expression
	phylogenetic_analysis
	sequence_alignment
	other

4.2.1 Modelo de Dados da Fase de Filtragem

A fase de filtragem inclui diferentes entidades com propósitos específicos. A tecnologia usada no sequenciamento de alto desempenho é representada pela entidade *sequencer*. Todas as SRS sequenciadas são representadas pela entidade *short_read* que armazena as sequências e as qualidades correspondentes a cada base. A entidade *filtering_process* descreve o processo de filtragem, onde cada instância da entidade *filtering_process* pode ser associada a diferentes parâmetros e seus valores através da entidade *filtering_parameter*. Os parâmetros usados são armazenados na entidade *parameter* que está relacionada com a entidade *filtering_parameter*. Além disso, as SRS têm diferentes tipos de qualidades de acordo com a tecnologia de sequenciador usada, o tipo de qualidade é representado pela entidade *quality_type*. As entidades *sample* e *organism* armazenam informações das amostras e do organismo da onde vem as amostras a serem sequenciadas. Enquanto isso, a entidade *project* representa os dados básicos do projeto de sequenciamento e está associado à entidade *sample* visto que num projeto são estudadas muitas amostras. Finalmente,

o autorelacionamento *filters* (veja figura 4.3) evidencia aquelas SRS que satisfazem o critério da filtragem, tais como porcentagem mínima de bases, onde cada um tenha um mínimo de qualidade, ou outros filtros de acordo com o objetivo do projeto. As SRS filtradas serão incluídas na próxima fase do *pipeline*.

A Figura 4.3 mostra o diagrama com os relacionamentos que existem no modelo Filtragem, composto pelas entidades: *organism, sample, project, sequencer, short_read, quality_type, filtering_process, filtering_parameter, e parameter*. A Tabela 4.3 apresenta os atributos de cada entidade desses modelos.

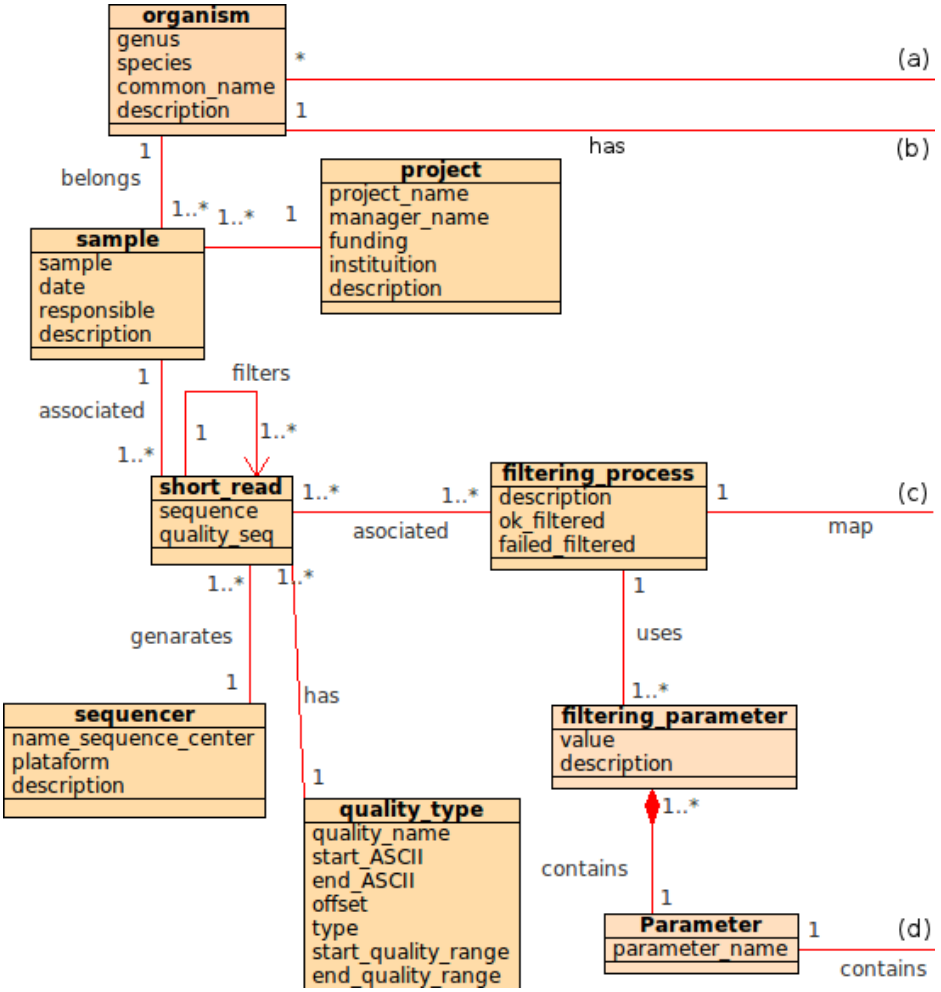

Figura 4.3: Diagrama de classes do modelo filtragem.

Tabela 4.3: Entidades e atributos do modelo filtragem

Entidade	**Nome atributos**	**Descrição atributos**
organism	*genus*	Gênero do organismo
	species	Espécie do organismo
	common_ name	Nome comum do organismo
	description	Descrição do organismo
sample	*sample*	Nome da amostra
	date	Data de preparação da amostra
	responsible	Pessoa encarregada
	description	Descrição
project	*project_ name*	Nome do projeto
	namager_ name	Pessoa encarregada do projeto
	funding	Fundação que financia o projeto
	instituition	Instituição a cargo do projeto
	description	Descrição do projeto
short_ read	*sequence*	sequência de bases
	quality_ seq	sequência de caráteres ASCII que representam a qualidade de cada base
sequencer	*name_ sequencer_ center*	Nome do centro de sequenciamento
	plataform	Tecnologia do sequenciador
	description	descrição
quality_ type	*quality_ name*	Nome do tipo de qualidade
	start_ ASCII	início dos caráteres ASCII
	end_ ASCII	Fim dos caráteres ASCII
	offset	Deslocamento
	type	Tipo de escore de qualidades (PHRED, Solexa, ...)
	start_ quality_ range	Início de qualidade
	end_ quality_ range	Fim de qualidade
filtering_ process	*description*	Descrição do processo de filtragem
filtering_ paramenter	*value*	Valor do parâmetro
	description	Descrição do valor usado
parameter	*parameter_ name*	Nome do parâmetro

4.2.2 Modelo de Dados da Fase de Mapeamento

Na fase de mapeamento (ver Figura 4.4) são usados diferentes genomas de referência (entidade *reference_ genome*) o que dependerá do tipo de organismo que está sendo estudado. Comumente é utilizado como o genoma de referência, o genoma de um organismo próximo bem anotado. Neste genoma de referência são mapeados as SRS filtradas (da fase de filtragem). A localização de cada SRS dentro do genoma de referência é representada pela entidade *mapping_ result*, que contém o sentido da fita, o começo e a posição final das SRS dentro do genoma de referência. Além disso, a entidade *reference_ genome* consiste de uma coleção de cromossomos que tem uma relação de composição com a entidade *chromosome* que representa todos os cromossomos que compõem o genoma de referência. Cada cromossomo pode ser considerado como uma sequência longa de DNA/RNA, o que por sua vez consiste de uma sequência (potencialmente sobrepostos) de fragmentos de sequências (entidade *chromosome_fragment*) com seus inícios e fins. No modelo da Figura 4.4, a entidade *mapping_ process* é associada às SRS filtradas ao genoma de referência e está relacionada com a entidade *mapping_ parameter* que representa os valores usados dos paramentos no processo de mapeamento.

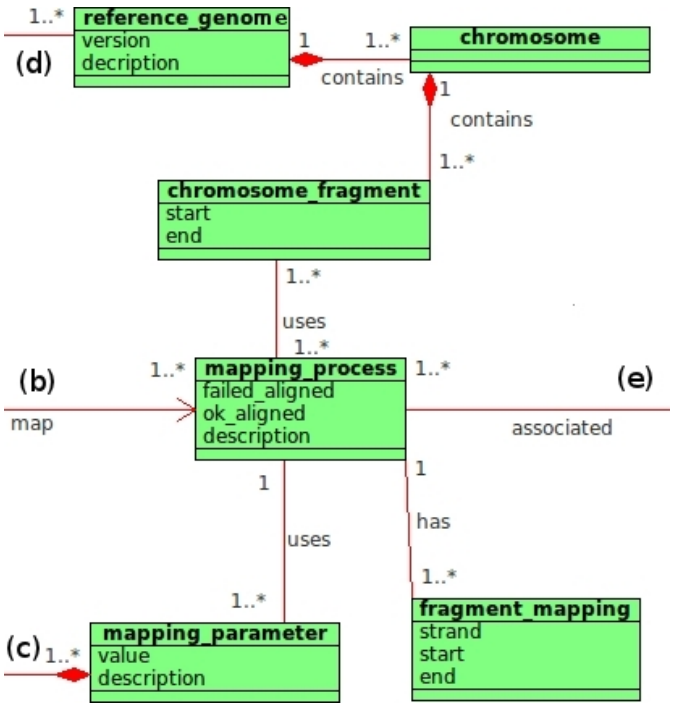

Figura 4.4: Diagrama de classes do modelo mapeamento.

A Figura 4.4 mostra o diagrama com os relacionamentos que existem no modelo Mapeamento, onde (b),(c), (d) e (e) unem as entidades *organism* da fase de filtragem à entidade *reference_ genome* da fase de mapeamento, a entidade *filtering_ process* da fase de filtragem à entidade *mapping_ process*, a entidade *parameter* da fase de filtragem à entidade *mapping_ parameter*, e a entidade *mapping_ process* à entidade *analysis_ process* da fase de análise respectivamente. A Tabela 4.4 apresenta a descrição de cada uma dessas entidades.

Tabela 4.4: Entidades e atributos do modelo mapeamento

Entidade	**Nome atributos**	**Descrição atributos**
Reference_ genome	*version*	Versão do genoma
	description	descrição
Chromosome	—	—
Chromosome_fragment	*start*	Começo do fragmento de sequência
	end	Fim do fragmento de sequência
Mapping_process	*failed_ mapped*	Número de SRS não mapeadas
	ok_ mapped	Número de SRS mapeadas com sucesso
	description	descrição
Mapping_ parameter	*value*	Valor de parâmetro
	description	descrição
mapping_ resul	*strand*	Sentido da cadeia (fita)
	start	Começo de onde foi mapeado a SRS
	end	Fim de onde foi mapeado a SRS

4.2.3 Modelo de Dados da Fase de Análise

Na fase de análise do *pipeline* de sequenciamento de alto desempenho, muitos tipos de análises podem ser desenvolvidos. O modelo desta fase é um modelo geral das principais análises que podem ser realizadas em um projeto de sequenciamento de alto desempenho. Neste contexto, a Figura 4.5 contém a entidade *database* que armazena informação sobre os bancos de dados usados para os diferentes processos de análises, assim como também, a entidade *analysis_ process*.

Outras entidades gerais são representadas no modelo e devem ser utilizadas de acordo com o tipo de análise realizada no projeto de sequenciamento, dentre essas tem-se as entidades: *sequence_ alignment* que representa os processos de alinhamento de sequências; *ncRNA_ identification* que representa a identificação de RNA não codificadores; *differential_ expresion* que representa o estudo de expressão diferencial; *phylogenetic_ analysis* que representa os processos de análises filogenéticas (identificação de genes homólogos, ortologos, e parologos), e *other_ analysis* que representa outros tipos de análises em específico que podem ser feitos. A Figura 4.5 apresenta esse modelo. A Tabela 4.5 apresenta a descrição de cada entidade.

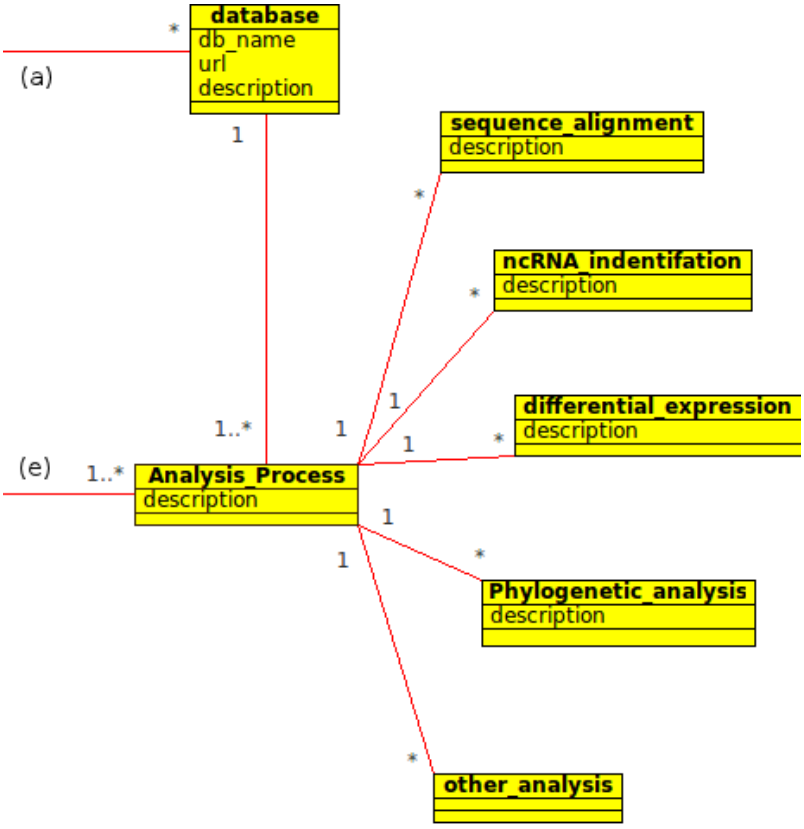

Figura 4.5: Diagrama de classes do modelo de análise.

Tabela 4.5: Entidades e atributos do modelo análise

Entidade	Nome atributos	Descrição atributos
database	*db_ name*	Nome do banco de dados usado
	url	Web site
	description	Descrição do banco de dados
analysis_ process	*description*	Descrição geral do processo de análise
ncRNA_ identification	*description*	Descrição do processo de identificação de RNA não codificadores
differentiall_ expression	*description*	Descrição do processo de expressão diferencial
phylogenetic_ analysis	*description*	Descrição do processo de análise filogenética
sequence_ alignment	*description*	Descrição do processo de alinhamento de sequências
other_ analyssis	—	—

Como a Tabela 4.5 apresenta, é possível ser realizados diferentes processos de análises. Neste trabalho de dissertação foi desenvolvido a análise de expressão diferencial

(*differential_ expression*).

4.3 Definição do Esquema Relacional do *Pipeline*

Para mostrar a viabilidade do modelo de dados conceitual desenvolvido na seção 4.2, é apresentado o esquema relacional gerado a partir do mapeamento do modelo conceitual. O esquema relacional está dividido em três partes seguindo o *pipeline* (geral) já definido (veja Figura 4.6).

O esquema relacional contempla algumas modificações inerente ao processo de mapeamento do modelo conceitual e do próprio esquema relacional. As tabelas resultam exclusivamente das entidades do modelo conceitual e das associações. Além disso, foi necessaria a criaçao de um novo tipo de dados para agrupar e armazenar milhões de SRS em um só elemento, já que a representação e consequentemente a inserção das SRS individualmente é custosa em termos de tempo e espaço dentro do SGBD, já que a criação dos metadados, índices e dados estatísticos fazem a inserção das SRS individualmente demoradas e aumentam o tamanho do banco de dados significativamente. Neste contexto, as SRS são agrupadas em conjuntos grandes o suficiente para serem armazenados diretamente no SGBD, porque os modernos SGBDs têm o tipo de dados BLOB que pode armazenar grandes quantidades de dados. Além disso, os SGBDs têm a capacidade e algoritmos de compressão de dados para o armazenamento eficiente de grandes volumes de informação. A Tabela 4.6 apresenta as tabelas que compõem o esquema do *pipeline*.

O esquema relacional está dividido em três esquemas. Estes esquemas representam as fases da filtragem, mapeamento e a análise. A Tabela 4.7 mostra para cada esquema, as tabelas que o compõe.

4.3.1 Esquema Relacional da Fase de Filtragem

A Figura 4.7 mostra o esquema relacional de dados utilizado no armazenamento da fase de filtragem. A tabela *filtering_ process* contém informação sobre os processos de filtragem tais como o número de SRS filtradas com sucesso e o número de SRS descartadas, sendo associado a um ou mais arquivos que contém SRS (tabela *short_ read*) através de uma tabela associada (*short_ read_ filtering*). A tabela *filtering_ process* está relacionada com os diferentes resultados inerentes a fase de filtragem. Esses resultados são armazenados na tabela *filtering_ result*. A tabela *filtering_ result* é resultado do auto relacionamento *filter* do modelo conceitual, já que, no processo de filtragem gera-se conjuntos grandes de SRS que passaram o processo de filtragem, no enquanto no modelo conceitual a filtragem é aplicada a nível de SRS. Consequentemente, com a tabela de resultados (*filtering_ result*) tem-se melhor controle dos resultados a nível de conjunto de SRS. Além disso, a tabela *filtering_ result* contém a coluna *data* criado para poder armazenar arquivos inteiros, *data* é definido usando o tipo de dados BLOB (*Binary Large Objects*). Por outro lado, a tabela *filtering_ process* está associada a diferentes parâmetros de filtragem que dependem dos diferentes filtros adotados armazenado na tabela *filtering_ parameter* que contém os parâmetros e os valores usados nas diferentes execuções para poder conseguir os resultados da tabela *filtering_ result*.

As tabelas *organism, sample, project, quality_ type, sequencer e parameter* não apresentam mudanças com respeito as suas correspondentes entidades do modelo conceitual.

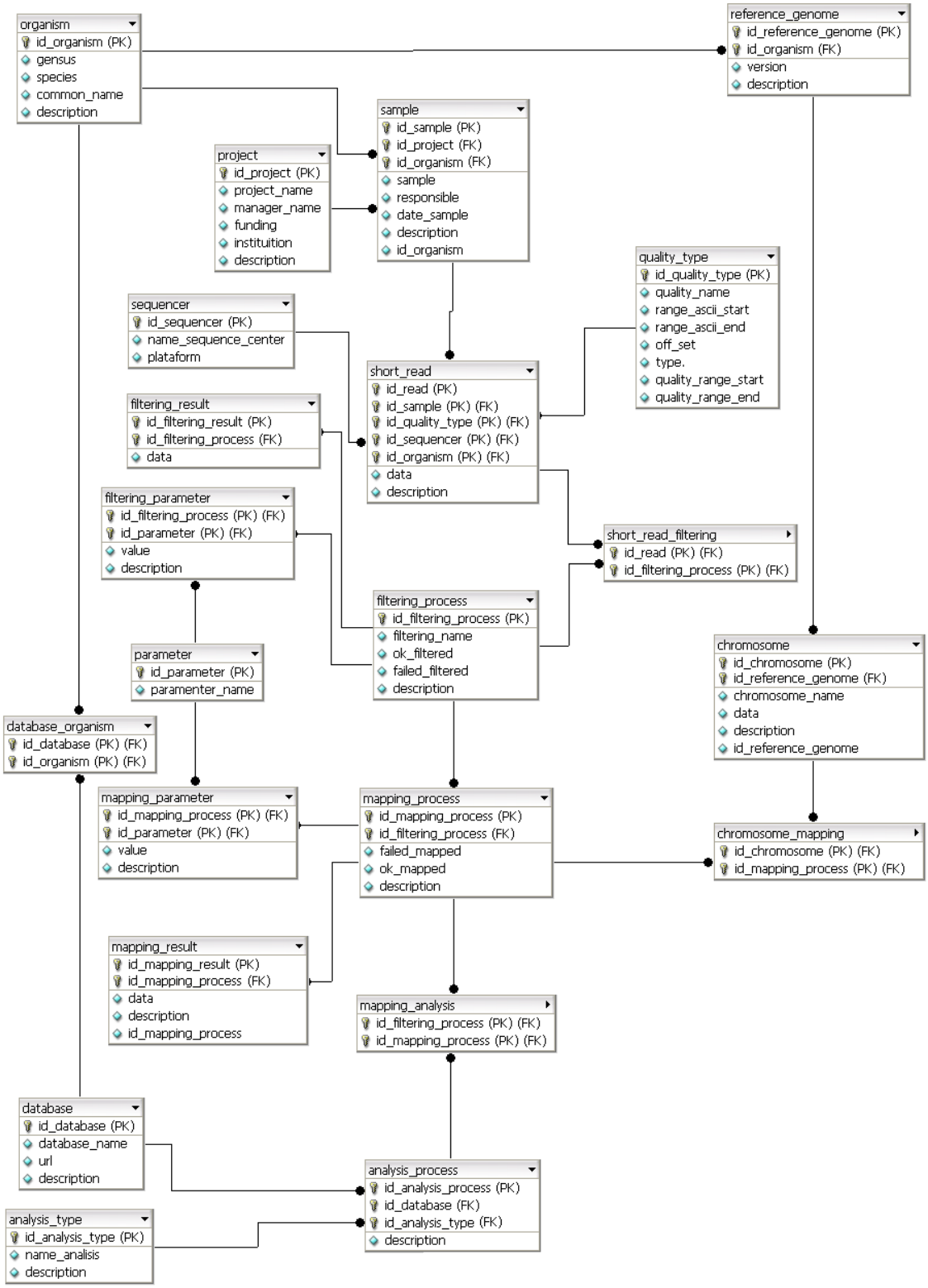

Figura 4.6: Esquema relacional do *pipeline* de sequenciamento de alto desempenho transcritômico. Ver diagrama ampliado no anexo II.

Tabela 4.6: Tabelas do esquema relacional do *pipeline*.

Nome tabela	**Descrição entidade**
organism	Organismos a serem estudados
sample	Amostras tiradas de algum organismo
project	Os projetos desenvolvidos
short_read	sequências de bases
sequencer	Sequenciadores
quality_type	Tipos de qualidades usados pelo sequenciador
filtering_process	Processo de filtragem das sequências
parameter	Parâmetros usados no processo
filtering_parameter	Valores dos parâmetros usados na filtragem
short_read_filtering	Tabela associativa entre as tabelas *short_read* e *filtering_process*
filtering_result	Resultado do processo de filtragem contemdo o conjunto de sequências de bases já filtradas
reference_genome	Genomas de referência bem anotados
chromosome	Cromossomos do genoma de referência
chromosome_mapping	Tabela associativa entre as tabelas *chromosome* e *mapping_process*
database_organism	Tabela associativa entre as tabelas *database* e *organism*
mapping_process	Processos de mapeamento
mapping_result	Conjunto de sequências mapeadas
mapping_parameter	Valores dos parâmetros usados no mapeamento
database	Bancos de dados usados
mapping_analysis	Tabela associativa entre as tabelas *mapping_process* e *analysis_process*

A Tabela 4.8 apresenta algumas tabelas, colunas e suas descrições (Anexo III para ver a tabela completa).

Tabela 4.7: Tabelas que compõem cada subesquema

Nome subesquema	**Nome tabela**
Filtragem	*Organism*
	sample
	project
	short_ read
	Sequencer
	quality_ type
	filtering_ process
	filtering_ parameter
	parameter
	short_ read_ filtering
	filtering_ result
Mapeamento	*Reference_ genome*
	chromosome_ mapping
	chromosome
	Mapping_ process
	Mapping_ parameter
	chromosome_ mapping
	mapping_ result
Análise	*database*
	database_ organism
	analysis_ process
	mapping_ analysis
	analysis_ type
	gene

4.3.2 Esquema Relacional da Fase de Mapeamento

A Figura 4.8 mostra o esquema relacional de dados utilizados no armazenamento da fase de mapeamento. Na tabela *mapping_ result*, a coluna *data* é de tipo BLOB (*Binary Large Objects*) onde arquivos resultado do mapeamento são armazenados. Analogamente, relatórios de resultados dos programas de mapeamento de SRS, tais como Bowtie e TopHat, são armazenados na tabela *mapping_ process*. Além disso, a tabela *mapping_ process* contém as colunas *failed_ mapped* e *ok_ mapped* que guardam o número de SRS mapeadas dentro do genoma de referência (tabela *reference_ genome*). O genoma de referência é composto de cromossomos (representado pela tabela *chromosome* contendo um arquivo por cada cromossomo armazenado dentro do campo *data* de tipo BLOB). Já, a tabela *chromosome_ mapping* é resultado da cardinalidade "muito para muitos" associado às tabelas *mapping_ process* e *chromosome*. Assim como na fase da filtragem, no processo de mapeamento são usados diferentes parâmetros e valores dos mesmos para cada execução do processo de mapeamento. Esses parâmetros e seus respectivos valores são armazenados na tabela *mapping_ parameter* que são detalhadas na Tabela 4.9 (Anexo IV para ver a tabela completa) para todas as tabelas envolvidas na fase de mapeamento.

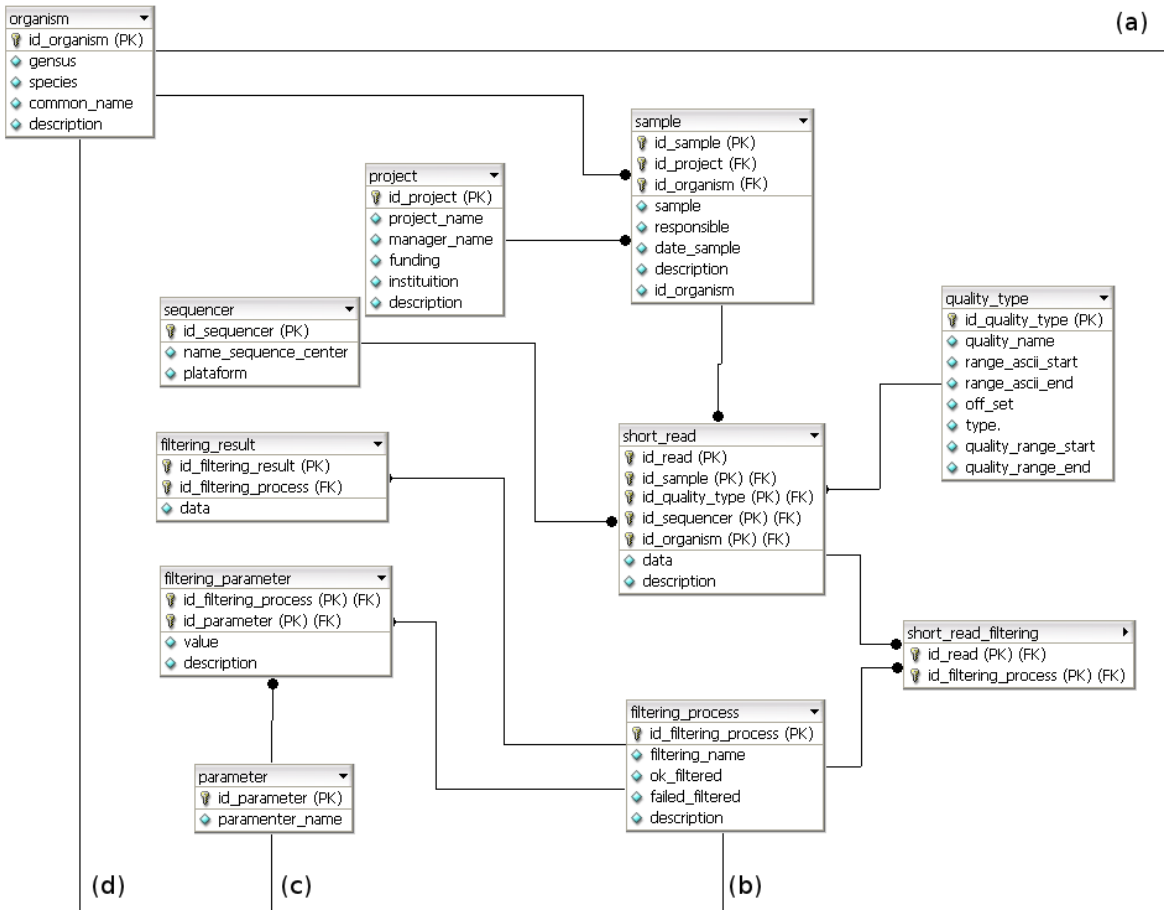

Figura 4.7: Esquema relacional da fase de filtragem.

4.3.3 Esquema Relacional da Fase de Análise

A Figura 4.9 mostra o esquema relacional de dados utilizados no armazenamento da fase de análise que no estudo de caso é a expressão diferencial. A tabela *mapping_ analysis* é uma tabela associada das tabelas *mapping_ process* e *analysis_ process* por causa da relação N:N entre as entidade *mapping_ process* e *analysis_ process* do modelo conceitual. A tabela *analysis_ process* armazena informação dos diferentes processos de análise. Além disso, a tabela *analysis_ type* armazena a especificação dos diferentes tipo de análises. Neste esquema pode-se adicionar outros esquemas que um processo de análise em especifico usa para desenvolver a análise.

A Tabela 4.10 explica detalhadamente cada coluna que compõe as tabelas envolvidas nesta fase.

Tabela 4.8: Tabelas e colunas do subesquema filtragem

Tabela	**Coluna**	**Descrição coluna**
organism	*id_ organism*	Identificador do organismo
	genus	Gênero do organismo
	species	Espécie do organismo
	common_ name	Nome comum do organismo
	description	Descrição do organismo
sample	*id_ sample*	Identificador da amostra
	sample	Nome da amostra
	date	Data de preparação da amostra
	responsible	Pessoa encarregada
	description	Descrição
project	*id_ project_ name*	Identificador do projeto
	project_ name	Nome do projeto
	namager_ name	Pessoa encarregada do projeto
	funding	Fundação que financia o projeto
	instituition	Instituição a cargo do projeto
	description	Descrição
short_ read	*id_ read*	Identificador do conjunto contendo as SRS
	id_ sample	Chave forânea da onde vem as SRS
	id_ quality_ type	Chave estrangeira da tabela *quality_ type*, indica o tipo de qualidade que das SRS
	id_ sequencer	Chave estrangeira da tabela *sequencer*, indica a tecnologia usada no sequenciamento
	data	Contém um conjunto de SRS
	description	descrição

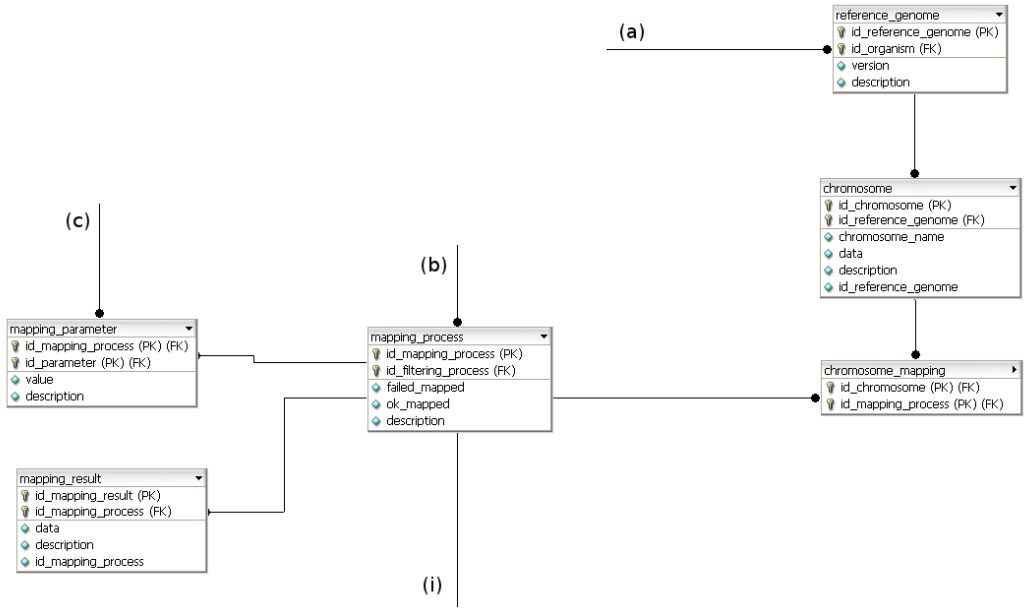

Figura 4.8: Esquema relacional da fase de mapeamento.

Tabela 4.9: Tabelas e colunas do subesquema mapeamento

Tabela	**Coluna**	**Descrição coluna**
reference_ genome	*id_ reference_ genome*	Identificador do genoma de referência
	id_ organism	Chave estrangeira da tabela *organism*, identifica o organismos
	version	Versão do genoma
	description	descrição
chromosome	*id_ chromosome*	Identificador do cromossomo
	id_ reference_ genome	Chave estrangeira da tabela *reference_ genome*, identifica o genoma de referência.
	chromosome_ name	Nome do cromossomo
	data	Contém a sequência de bases do cromossomo
	description	descrição
chromosome_ mapping	*id_ mapping_ process*	Chave estrangeira da tabela *mapping_ process*, identifica o processo de mapeamento
	id_ chromosome	Chave estrangeira da tabela *chromosome*, associa o cromossomo com o processo de mapeamento

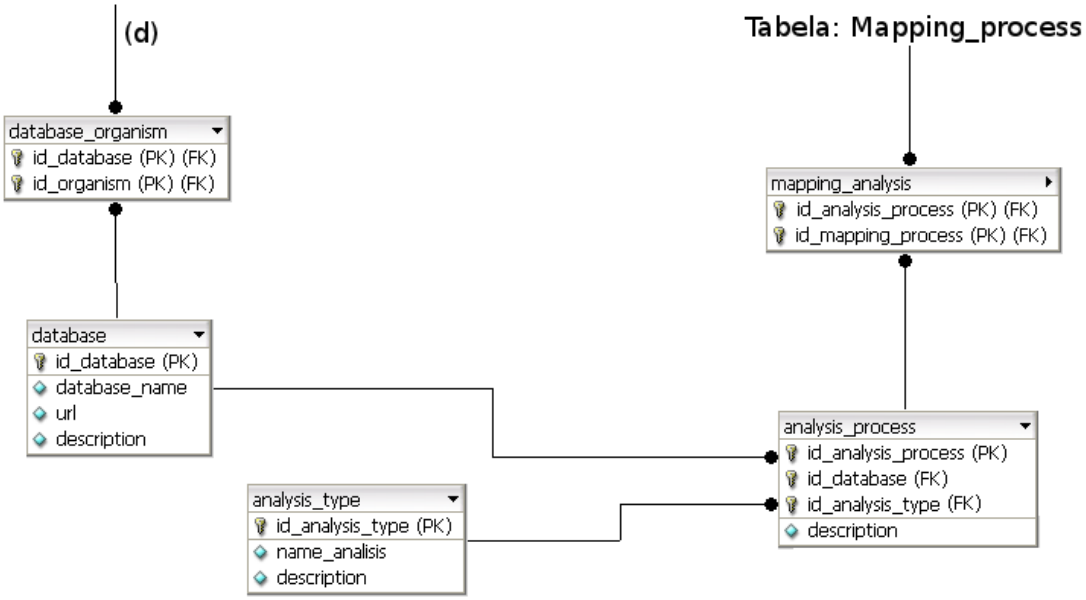

Figura 4.9: Esquema relacional da fase de análise.

Tabela 4.10: Tabelas e Colunas do subesquema a fase de análise - Expressão.

Tabela	**Coluna**	**Descrição coluna**
analysis_process	*id_analysis_process*	Identificador do processo de análise
	id_database	Chave estrangeira da tabela *database*, identifica o banco de dados usado
	id_analysis_type	Chave estrangeira da tabela *analysis_type*, identifica o tipo de análise
	description	descrição da análise
analysis_type	*id_analysis_type*	Identificador do tipo de análise
	name_analysis	Nome da análise
	description	descrição do tipo de análise
database	*db_name*	Nome do banco de dados usado
	url	Sitio web
	description	Descrição
database_organism	*id_database*	Chave estrangeira da tabela *database*, identifica o banco de dados
	id_organism	Chave estrangeira da tabela *organism*, identifica o organismo
mapping_analysis	*id_mapping_process*	Chave foránea da tabela *mapping_process*, identifica o processo de mapeamento
	id_analysis_process	Chave foránea da tabela *analysis_process*, identifica o processo de análise

Capítulo 5

Estudo de Caso

No presente capítulo apresenta-se dois estudos de caso com o objetivo de validar a proposta de modelagem e implementação apresentada no Capítulo 4. Na Seção 5.1 é apresentada uma visão geral dos estudos de casos implementados. Na Seção 5.2 são estudadas as diferentes tecnologias usadas nos estudos de caso. Na Seção 5.3 é definido o *pipeline* com as aplicações da bioinformática incluindo as fases: filtragem, mapeamento e análise. Na Seção 5.4 são apresentados os resultados experimentais. Na Seção 5.5 apresenta-se a discussão dos modelos e dos resultados alcançados. Na seção 5.6 apresentam-se os trabalhos publicados.

5.1 Visão Geral do Estudo de Caso

Para a avaliação do modelo conceitual proposto, foram desenvolvidos dois estudos de caso utilizando dados gerados no Departamento de Genética Humana da Universidade de Chicago dos Estados Unidos da América publicado em 2008 [55] e dados gerados pelo laboratório YEO LAB da Universidade de Califórnia dos Estados Unidos da América publicado em 2008. Os dados gerados por esses laboratórios são provenientes do sequênciamento de alto desempenho onde foram usados os sequênciadores Illumina GA II e Illumina GA I, respectivamente.

Em primeiro lugar, o laboratório do Departamento de Genética Humana da Universidade de Chicago realizou o sequênciamento de amostras de células de rim e fígado para identificar a expressão diferencial de genes em comparação com tecnologias de arranjos [56] existentes. Neste contexto, o objetivo desse trabalho foi comparar a capacidade de identificar genes diferencialmente expressos entre duas abordagens diferentes: sequênciamento de alto desempenho e tecnologia de arranjos. O sequênciamento das amostras de cDNA de rim produziu 72 987 691 SRS e a amostra de fígado 72 126 823 SRS . O resultado do sequênciamento é um conjunto de arquivos FASTQ contendo SRS de 36 pares de bases de comprimento com as suas qualidades associadas a cada base.

No segundo caso, o laboratório YEO LAB da Universidade de Califórnia desenvolveu um estudo de análise transcritômico com células de câncer para detectar transcritos e isoformas de mRNA. O objetivo deste trabalho foi comparar os dados do sequênciamento de células de câncer de próstata LNcap que receberam tratamento com hormônios de andrógeno [57] através de uma análise quantitativa da expressão diferencial de genes. O sequênciamento de células de câncer de próstata LNcap com e sem tratamento produziu

10 109 398 SRS para amostras tratadas e 7 156 324 SRS para amostras não tratadas. O resultado do sequênciamento é um conjunto de arquivos FASTA contendo SRS de 36 pares de base de comprimento. As SRS deste sequênciamento não apresentam as sequências das qualidades correspondentes a cada base, por isso o tratamento destes dados tem algumas peculiaridades que são descritas nas próximas seções.

5.2 Arquitetura Abstrata do *Pipeline*

Para a execução do *pipeline* é necessário definí-lo e configurá-lo. Sendo assim, a primeira etapa foi a definição dos programas a serem utilizados, assim como a sua configuração em cada fase do *pipeline*. O esquema apresentado na Figura 5.1 mostra o funcionamento do *pipeline*.

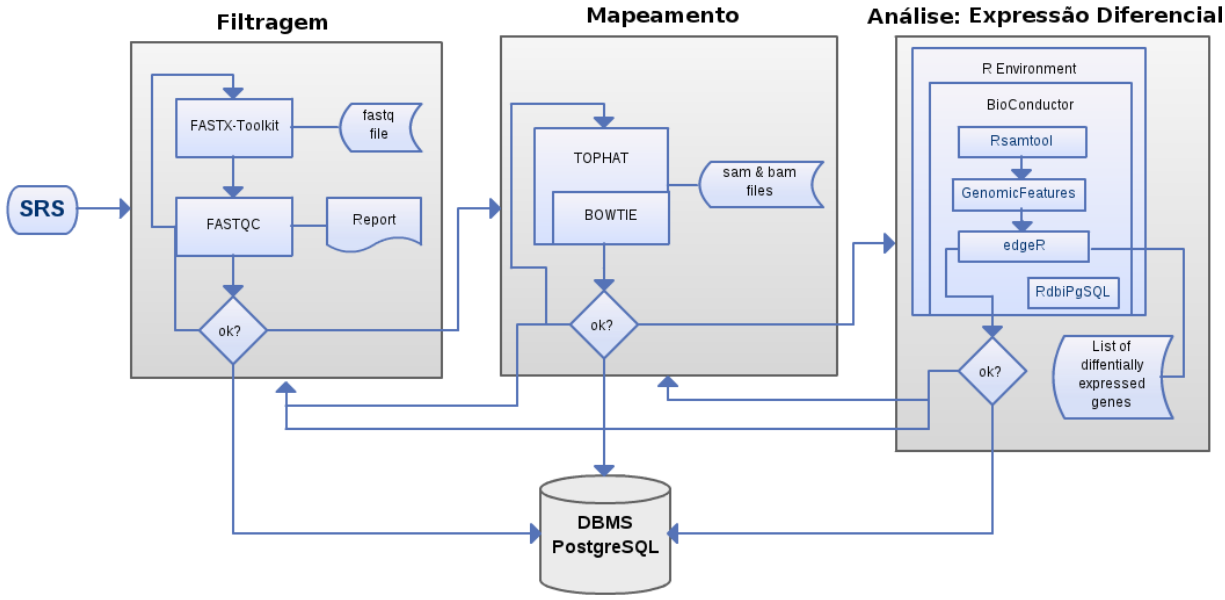

Figura 5.1: Visão geral do *pipeline* de análise para sequênciamento de alto desempenho transcritômico usado como estudo de caso.

Como dito na seção anterior, as SRS utilizadas no estudo de caso têm o formato FASTQ [58]. Em linhas gerais o formato FASTQ é composto pelas cadeias de sequências de bases e as sequências de qualidades associadas a cada base. Este tipo de arquivo armazena informação gerada pelos sequênciador *Illumina* em formato texto (Ver Anexo VII). Os arquivos FASTQ tem um grande volume de dados, chegando ao tamanho de mais de 10GB de dados nos nossos estudos de caso.

Na fase de filtragem do *pipeline*, foram usados os pacotes FASTX-*Toolkit* e o pacote FASTQC. O FASTX-*toolkit* [59] é uma coleção de ferramentas que fornece pré-processamento de arquivos FASTA e FASTQ. Entre as principais caraterísticas tem-se a conversão do formato FASTQ a FASTA, remoção de *barcodes* de sequências, remoção de adaptadores de sequências, filtragem de sequências baseadas na qualidade, entre outras. O FastQC [60] é uma aplicação java que gera um relatório de controle de qualidade dos

dados de sequenciamento de alto desempenho com o objetivo de detectar problemas que se originam tanto no sequenciador ou no material usado no sequenciamento de alto desempenho. Esta ferramenta tem como entrada arquivos BAM, SAM e FASTQ, produzindo em sua saída figuras e relatórios da qualidade dos dados.

Sendo assim, no nosso *pipepiline* foi usado o pacote FASTX-*Toolkit* para eliminar as SRS de baixa qualidade, e os pacotes FASTQC para avaliar se os resultados alcançados foram aceitáveis através de informes estatísticos. Segundo os resultados alcançados pode-se continuar com a próxima fase ou realizar outro processo de filtragem. Esta fase da filtragem é de suma importância para assegurar que a fase seguinte do *pipeline* use só sequências com qualidade aceitável.

Uma vez que a fase de filtragem foi completada, o processo de mapeamento começa usando o programa TopHat. O TopHat [61] implementa um algoritmo de mapeamento de SRS eficiente projetado para alinhar SRS que vem de um sequenciamento de alto desempenho. O TopHat encontra junções mapeando as SRS em duas fases. Na primeira fase, são mapeadas todas as SRS no genoma de referência usando Bowtie [62] que usa índices para acelerar o procedimento de busca e diminuir o custo de memória associado a procura das sequências no genoma de referência. Esta técnica usada pelo Bowtie consiste em concatenar todo o genoma de referência em uma única *string* e realizar uma transformação de Burrows-Wheeler para construir um índices do genoma de referência. O programa então procede realizando o mapeamento de um caracter da SRS por vez, até alinhar todas as SRS. Se isso não for possível, o programa volta atrás e realiza a substituição de um caracter, uma opção permite controlar o número máximo de substituições de carateres permitidas. Todas as SRS que não foram mapeadas no genoma são separadas como SRS não mapeadas inicialmente. Depois, as SRS não mapeadas são divididas em segmentos menores e mapeadas individualmente. Dessa forma, amplia-se as probabilidades de ser mapeadas no genoma de referência.

O programa R foi escolhido para implementar a análise de dados. O R [63] é um ambiente de *software* livre para computação estatística. Trabalha sobre diferentes plataformas: UNIX, Windows e MacOS. Uma das principais vantagens do R é a facilidade de projetar *plots* de qualidade, incluindo símbolos e fórmulas matemáticas, quando necessárias. Outra importante vantagem é a facilidade de inclusão de diferentes aplicativos tal como o projeto BioConductor [64] que fornece ferramentas para as análises e compreensão de dados de sequenciamento de alto desempenho.

Entre os diferentes pacotes oferecidos pelo projeto BioConductor tem-se o pacote Rsamtool [65] que traz as funcionalidades do samtool através dos métodos scanBAM e BAM Views. O método scanBAM é altamente parametrizado de modo que muitos detalhes de acesso e de filtragem de arquivos BAM contendo SRS podem ser controlados através do R. O método BAM Views permite a leitura e gerenciamento dos dados no R; SRS mapeadas podem ser importadas, e visualizadas eficientemente para grandes coleções de dados. O pacote de *GenomicFeatures* [11], é um conjunto de ferramentas e métodos para fazer e manipular anotações de transcritos. Com estas ferramentas o usuário pode facilmente baixar as localizações genômicas dos transcritos, exons e CDS de um dado organismo. Esta informação é armazenada em um banco de dados local que mantém o controle da relação entre os trancritos, exons CDS e genes. O *GenomicFeatures* também fornece métodos flexíveis para extrair as caraterísticas desejadas em um formato conveniente. O Pacote edgeR (*Empirical analysis of Digital Gene Expression data in R*) [66], usa

métodos de Bayes empírico e distribuição binomial negativa para as análises de expressão diferencial de sequênciamento de alto desempenho. EdgeR é projetado para a análise de dados baseado na contagem da replicação de dados. Finalmente, o pacote RdbiPgSQL [67] fornece métodos para acessar dados armazenados em tabelas do SGBD PostgreSQL usando o ambiente R.

O Sistema Gerenciador de Banco de Dados

Os sistemas de bancos de dados são projetados para administrar grandes volumes de informações sobre uma determinada aplicação, provendo um ambiente que seja adequado e eficiente para o armazenamento e a recuperação das mesmas [35]. Um dos principais benefícios de um sistema de banco de dados é proporcionar uma visão abstrata dos dados. Uma vez que a maioria dos usuários de bancos de dados não é especialista em computação, omite-se deles a complexidade da estrutura interna dos bancos de dados, graças a diversos níveis de abstração que simplificam a interação do usuário com o sistema [35].

De uma maneira geral, pode-se dizer que um sistema de banco de dados é constituído por um conjunto de programas e/ou aplicações; estes, por sua vez, estão associados a um conjunto de dados por intermédio de um SGBD [35]. Neste contexto, a integração dos dados em banco de dados, acessados, tanto pelos programas como por consultas, por meio de uma linguagem de alto nível foi possível através do SGBD. O SGBD, parte integrante de um sistema de banco de dados (veja Figura 5.2), é um software que ajuda os usuários a criar, armazenar e processar dados para diversas aplicações [36, 35]. O SGBD é o responsável pelo controle de acesso aos dados, ou seja, é ele que gerencia os privilégios de cada um dos usuários, e libera, ou não, o acesso aos dados, geralmente por meio de um sistema de acesso a usuários. Além disso, os SGBDs devem garantir as seguintes características: Controle de Transações, Garantia da Integridade, Garantia de Segurança [35].

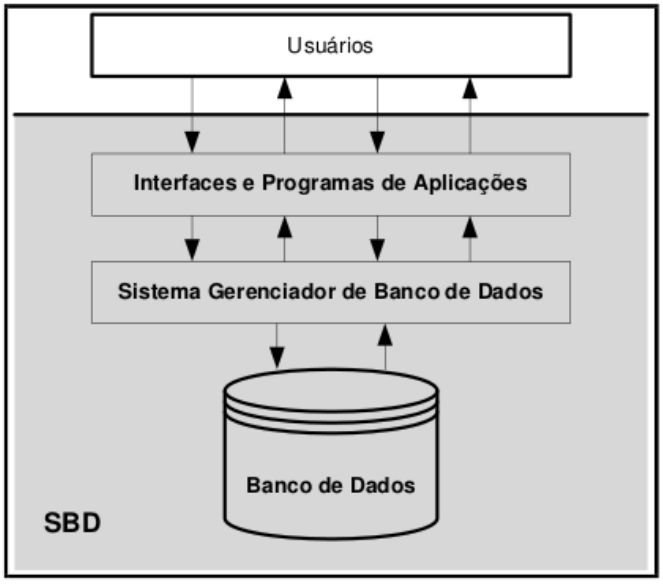

Figura 5.2: Representação simplificada de um Sistema de Banco de Dados.

Nesta pesquisa, optou-se pelo SGBD PostgreSQL que possui como ambiente nativo a plataforma Unix, sendo também compatível com a plataforma aberta Linux que é bastante usada na área de bioinformática. Outra característica que possui é uma interface gráfica através de um cliente no ambiente MS Windows, bem como nas plataformas Linux e Unix. Além disso, realizou-se análises de tempo gasto e de espaço na inserção de grandes volumes de dados no SGBD MySQL comparado com o SGBD PostgreSQL, onde o SGBD PostgreSQL obteve melhores resultados. Estes resultados influenciaram na escolha do PostgreSQL em relação ao MySQL no nosso estudo de caso.

O PostgreSQL conta uma extensa comunidade que suporte as diferentes organizações acadêmicas, corporativas e de pesquisa. PosgreSQL é líder em tecnologia e é conhecido como o SGBD de código aberto mais avançado do mundo. Ele tem excelente desempenho, alta segurança, é rico em funcionalidades, simples de usar, aprender e gerenciar.

O PostgreSQL é um SGBD objeto-relacional. Uma das suas principais vantagens é possuir recursos comuns a bancos de dados de grande porte. Além disso, trata-se de um banco de dados de alta versatilidade, seguro, com uma documentação atualizada e extensa, e gratuito.

O PostgreSQL assim como a maioria dos SGBDs relacionais oferecem mecanismos para manipular os dados através de linguagens textuais. Estas, por sua vez, são derivadas do SQL (*Structured Query Language*). Esta linguagem implementa mecanismos para atualizar e consultar os dados, e também mecanismos para expressar restrições de integridade dentro do SGBD [68].

Métricas Usadas no Estudo de Caso

A especificação da medida de avaliação é utilizada no processo de comparação relativa entre a abordagem SGBD e sistemas de arquivos. A medida de avaliação foi o espaço economizado (EE) pela abordagem SGBD em relação a uma abordagem usando sistemas de arquivos. A seguir, descrevemos a medida de avaliação de espaço economizado que é necessário para entender a avaliação no armazenamento.

A definição de espaço economizado (EE) é a redução do tamanho relativo ao tamanho descompactado [69]. Esta definição é apresentada na equação 5.1; enquanto o equivalente para nosso estudo de caso é apresentado na equação 5.2 (usada pelas Tabelas 5.1 e 5.2).

$$EE = 1 - \frac{Tamanho\ Compactado}{Tamanho\ Original}; \tag{5.1}$$

$$EE = 1 - \frac{Tamanho\ em\ SGBD}{Tamanho\ em\ Sistema\ de\ Arquivos}; \tag{5.2}$$

O tamanho em SGBD é o resultado depois que os dados são armazenados no SGBD e o tamanho em sistema de arquivos é o tamanho dos dados no formato original sem nenhum tipo de compactação.

Por outro lado, usou-se como métrica o tempo gasto pela inserção e exportação dos dados gerados nas diferentes fase comparada com o tempo gasto pelo processo de filtragem, mapeamento e análise respectivamente. Esta métrica é usada para mostrar a porcentagem de tempo usada no processo de inserção e exportação no SGBD em relação do tempo gasto no processo envolvido (filtragem, mapeamento e análise). A equação 5.3 apresenta esta definição e usada nas tabelas 5.3 e 5.4.

$$\%\ Tempo\ Gasto = \frac{Tempo\ de\ inserir/exportar\ no\ SGBD}{Tempo\ do\ processo} * 100\%; \qquad (5.3)$$

5.3 Discussão e Análises dos Resultados Experimentais do *Pipeline*

Após a execução do *pipeline* definido anteriormente o esquema relacional resultante incluí novas tabelas (o Anexo VI mostra o esquema geral resultante). O esquema da fase de análise é o único que sofre mudanças, onde são adicionadas quatro novas tabelas de resultados (*gene_result*, *transcript_result*, *exon_result* e *cds_result*) da expressão diferencial por transcritos, exons ou CDSs respectivamente. Estas tabelas de resultados contém o número de vezes que cada SRS foi mapeada no genoma de referência. Além disso são adicionadas outras tabelas que compõem o esquema de transcritos (TranscriptDB) que é gerado pelo pacote *GenomicFeatures* do programa R que fornece uma forma de recuperar, armazenar, e consultar recursos como exons, transcritos, e sequências codificadoras de muitos organismos de referência. A Figura 5.3 mostra o esquema da fase de análise após a execução do pipeline.

No Anexo V explica-se detalhadamente cada coluna que compõe algumas tabelas envolvidas neste novo esquema para a fase de análise. O esquema de transcritos armazena metadados de transcrição (informação das entidades envolvidas no processo de trancrição) que gerenciam localizações genômicas e as relações entre transcritos, exons e sequências codificadoras de proteínas [54]. Este esquema contém anotações de transcritos que estão relacionados entre si. A tabela *gene* está associada com a tabela *transcript* que armazena os transcritos de cada gene. De igual forma, as tabelas *exon* e *cds* que armazenam as regiões de exon e as regiões de sequências codificadoras de cada gene. No entanto, a tabela *chrominfo* mantém informação dos cromossomos envolvidos.

Para avaliarmos o modelo conceitual e seu respectivo esquema relacional, foram realizadas algumas análises que envolvem: (i) a viabilidade de criação de um modelo conceitual, comparando-o com modelos que já existem na literatura; (ii) a eficiência em termos de armazenamento de dados quando comparado com um sistema de arquivo; (iii) uma comparação entre o tempo de processamento ao se armazenar os dados envolvidos no *pipeline* dentro de um SGBD, em relação ao tempo de execução do *pipeline*, comparado com um sistema de arquivos.

Nas próximas subseções cada um desses temas é abordado.

5.3.1 Análises Sobre o Modelo Conceitual

O modelo de dados apresentado neste trabalho tem o objetivo de representar dados gerados pelos sequenciadores de alto desempenho no intuito de armazenar e administrar grandes volumes de dados baseada na modelagem orientada a objetos. Similarmente a outros trabalhos da literatura, a nossa abordagem faz uso do MOO e a UML para representar dados complexos da bioinformática.

Na revisão da literatura, foi possível verificar que a maioria de modelos para dados biológicos existentes estão mais interessados em representar os conceitos da biologia molecular, mas não os processos envolvidos tais como a filtragem de dados, mapeamento de

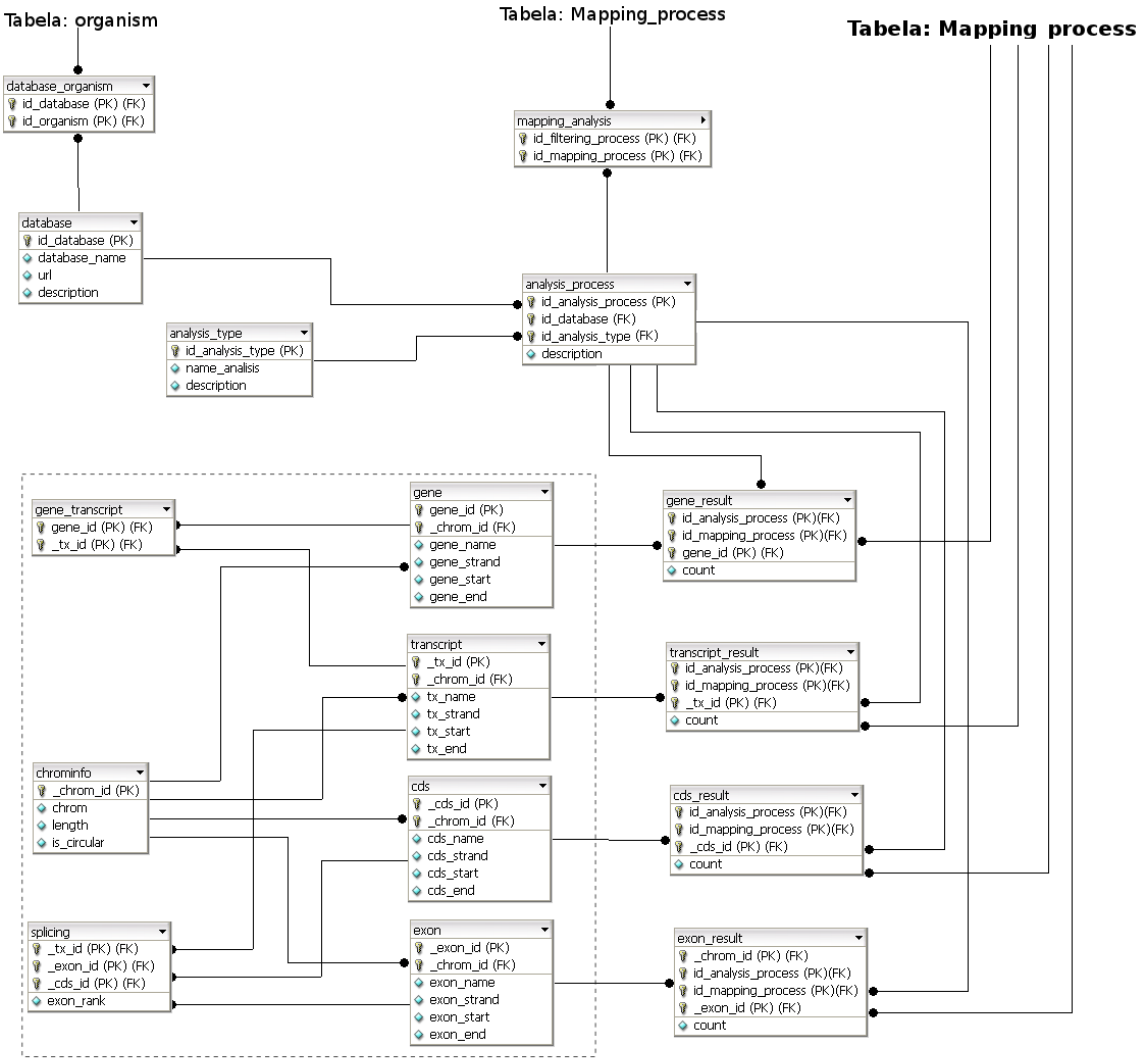

Figura 5.3: Esquema relacional da fase de análise – expressão diferencial. As linhas ponteadas de cor cinza delimita o esquema *TranscriptDB* gerado pelo pacote *GenomeFeatures*

sequências e as diferentes análises. Para começar, Paton et al. (2000) [5] e Bornberg-Bauer e Paton (2002) [46] representam conceitos e fenômenos da biologia molecular para sequências genômicas e proteicas. Elmasri et al. (2006) [6] e Macedo et at. (2007) [8] são trabalhos mais recentes também interessados na representação de dados biológicos. Estes trabalhos acrescentam algumas funcionalidades especiais para dessa forma acomodar o modelo EER (*Enhanced Entity Relationship*) e o MOO (Modelo Orientado a Objetos) representando de melhor forma conceitos da biologia molecular.

Na abordagem de Busch e Wedeman (2009) [7] é possível cumprir com os requisitos interoperabilidade e flexibilidade para o domínio da biologia molecular. Isto graças à definição de um modelo dinâmico. Enquanto que a abordagem que propomos está interessada na representação e organização de resultados parciais das análises ao longo do desenvolvimento de um *pipeline*, tais como controle de qualidade, mapeamento de SRS (*Short Read Sequences*) e a identificação da expressão diferencial de genes.

Os esquemas relacionais de banco de dados encontrados na literatura, tais como, o CHADO [10] e GUS [49] são esquemas relacionais genéricos para poder tratar dados da biologia molecular. Estes esquemas podem dar suporte a uma gama ampla de projetos por serem genéricos demais, contendo centenas de tabelas, fazendo destes esquemas muito complexos e difíceis de usar. No entanto, o modelo que propomos tenta integrar os dados gerados nas diferentes fases de um *pipeline* de uma forma simples e intuitiva para que os usuários possam usá-los para implementar sistemas de informação que administrem o *pipeline* todo de um sequenciamento de alto desempenho transcriptômico.

Uma vez que nosso modelo de dados está divido em três fases, a interoperabilidade entre elas é claramente visível, dessa forma o deslocamento entre as fases através do *pipeline* é mais fácil, pois, é muito importante poder fazer consultas e conseguir informação da fase atual, anterior ou próxima. Isto significa que podem ser extraídos diferentes tipos de informação desde o início da fase de filtragem até o final da fase de análise (em nosso caso, expressão diferencial de genes) através de diferentes consultas sobre o modelo. Por exemplo, o biólogo poderia questionar se os parâmetros que foram usados para conseguir o melhor resultado entre todos os resultados da fase de filtragem ou saber qual foi o melhor mapeamento para os dados da célula de rim. Para isto, pode-se juntar as tabelas *filtering_process, filtering_result, filtering_parameter e parameter* (ver Figura 4.2). Portanto, o esquema resultante do modelo proposto permite responder diferentes consultas sobre o processamento do *pipeline*, uma vez que o esquema relaciona as tabelas de processos: filtragem, mapeamento e análise (expressão diferencial).

5.3.2 Comparação da Eficiência no Armazenamento de Dados

Uma das preocupações no uso de um SGBD em relação ao sistema de aquivos era o aumento de espaço armazenado que essa tecnologia podia trazer. Por esse motivo uma das medidas de avaliação da proposta desta dissertação foi a eficiência no armazenamento em SGBD e o tempo gasto no armazenamento comparado com o tempo gasto no processo todo. A avaliação no armazenamento foi realizada comparando os dados armazenado em sistema de arquivos e os mesmos dados num SGBD. Os resultados obtidos levaram em consideração o espaço economizado sobre os dados. Além disso, a avaliação no tempo gasto no processamento dos dados em relação ao tempo gasto no armazenamento num SGBD. Foram usadas tabelas com tipo de dados heterogêneos e de fontes distintas, possibilitando assim uma melhor avaliação dos resultados. Os experimentos foram realizados num servidor HP (8 Intel(R) Xeron(R) de 8 CPUs de 2.13GHz, 22.66GB de 1333 MHz de memória RAM, 1 HD de 264GB SCSI) sobre o sistema operacional Linux Server Ubuntu/Linaro 4.4.4-14.

O sistema de arquivos é comumente adotado nos projetos transcritômicos e genômicos. Uma das vantagens dos arquivos é a facilidade da implementação e rápida execução quando comparados com os SGBDs. Uma vez que nosso trabalho está focado no armazenamento dos dados ao longo da execução do *pipeline*, como a entrada de dados é muito grande e usada com pouca frequência, o armazenamento eficiente terá grande impacto sobre o *pipeline*, quando comparado com o desempenho da execução. Nesse sentido, medimos a eficiência do armazenamento para dados armazenados em arquivos e no SGBD PostgreSQL que foi usado nos estudos de caso.

Os SGBDs modernos, entre os quais se incluem o PostgreSQL implementam o algoritmo de compressão, no PostgreSQL é utilizado o TOAST (*The Oversized-Attribute Storage Technique*) [70]. A compressão TOAST é habilitada automaticamente para todos os tipos de dados que contenham cadeias de caracteres e superam o tamanho de 2 KB. Uma vez superado o valor de 2KB por um atributo de alguma tabela, esse dado é armazenado em um tipo "*extension room*"(tabelas "TOAST") da tabela usada para armazenar (no sentido do tamanho dos dados) atributos com valores muito grandes que não cabem em páginas de dados normais (como textos longos) [70]. Além disto, os arquivos de tamanho muito grande são armazenados no tipo de dados BLOB.

A Tabela 5.1 mostra o tamanho total de espaço em disco para armazenar os dados do genoma de referência e os dados gerados pelo pacote GenomicFeature (banco de dados de transcritos), tanto para sistema de arquivos como para SGBD. No caso do genoma de referência, os dados (arquivos) são armazenado em colunas de tipo BLOB. Uma vez que esses dados são grandes demais, o algoritmo de compressão interna TOAST implementado pelo PostgreSQL é ativado, dessa forma obtendo uma taxa de economia de espaço de 51,1% para os dados do genoma de referência. No entanto, os dados do *TranscriptDB* alcançaram uma taxa de espaço economizado negativo de -195,7% o que significa que o tamanho dos dados no SGBD aumentaram de tamanho em um porcentagem de 195,7% do tamanho original. Uma vez que os dados do *TranscripDB* são pequenos demais para que o algoritmo de compressão TOAST seja aplicado e o aumento de dados como a criação de índices e/ou tabelas de índices associadas a cada inserção de dados muito pequenos; fazem que os dados originais do *TranscripDB* (dados gerados pelo pacote *GenomicFeatures*) aumentem de tamanho no SGBD. O espaço economizado total na Tabela 5.1 é 45,38%. Este resultado é consequência do volume maior dos dados do genoma de referência comparado ao volume dos dados do *TranscriptDB*. Ainda que o espaço economizado do *TranscriptDB* tenha sido negativo.

Tabela 5.1: Armazenamento para o genoma de referência e dados do *TranscriptDB*

	Sistema de Arquivos (MB)	**Esquema SGBD (MB)**	**Espaço Economizado (%)**
Genoma de referencia	2.745,0	1.343,0	51,1
Dados do *TranscriptDB*	64,6	191,0	-195,7
Total	2.809,6	1.534	45,38

Os melhores resultados em taxa de espaço economizado foram os apresentados na Tabela 5.2. A Tabela 5.2 contém valores relacionados com as fases filtragem, mapeamento e expressão diferencial na qual o volume dos dados é muito grande. A predominância de dados do tipo texto fez possível atingir o valor de espaço economizado total de 39,3%, ainda tendo obtido resultados negativos na fase de mapeamento onde os dados são binários e na fase de análise onde os dado são pequenos demais para que o TOAST seja ativado. Este resultado mostra que ainda tendo resultados negativos, estes não tem grande impacto no resultado final, já que os dados que obtiveram resultados negativos são pequenos comparados aos dados que obtiveram resultados positivos e ao volume total dos dados.

Na Tabela 5.2 o espaço economizado para as SRS de rim e fígado foi de 57.9% e 48.2% para dados de células de câncer de próstata LNCaP. Para as SRS filtradas foi de 54,8% e 48,8% respectivamente. A principal razão para obter esses valores é o fato de que arquivos FASTQ podem ser comprimidos eficientemente, já que são de tipo texto. Contudo, foram obtidos resultados negativos com arquivos que não são formados por cadeias de caracteres. Por exemplo, os arquivos BAM obtiveram uma taxa de espaço economizado negativa e não foram comprimidos pelo TOAST. Entretanto, os dados da expressão diferencial de genes obtiveram um valor de espaço economizado negativo, por não serem suficientemente grande para acionar o algoritmo de compressão TOAST. Em geral, os resultados de ambos estudos de casos (rim/fígado e câncer de próstata LNCaP) foram similares. Contudo, pode-se notar que as análises para dados de células de rim e fígado alcançaram melhor desempenho de armazenamento quando comparado aos dados de câncer de próstata LNCaP, e isto pode ser explicado pelo volume de dados de cada caso, porque os dados de rim e fígado são significativamente maiores que os dados de células de câncer de próstata LNCaP.

Nos casos onde se obtiveram valores negativos, o volume dos dados são significativamente menores, como podem-se ver nas Tabelas 5.1 e 5.2. Porém como o volume de dados nesses exemplos é pequeno em relação ao volume total no *pipeline*, esses valores negativos tiveram pouca influência no resultado geral. Os resultados mostraram também que quanto maior o tamanho dos dados de tipo texto, maior será a valor de espaço economizado devido ao algoritmo de compressão TOAST.

Tabela 5.2: Comparação de eficiência no armazenamento de dados de células de Rim/fígado e células de câncer de próstata LNCaP.

	Sistema de Arquivos (MB)		**Esquema SGBD (MB)**		**Espaço Economizado (%)**	
	Rim/ fígado	Câncer CNcap	Rim/ fígado	Câncer CNcap	Rim/ fígado	Câncer CNcap
SRS	35.691,5	843,9	15.023,0	437,0	57,9	48,2
SRS filtradas	30.176,4	843,9	13.629.0	432,0	54,8	48,8
Mapeamento	2.784,3	139,0	3.758.0	231,0	-35,0	-66,2
Dados da expressão diferencial de genes	2,4	2,2	10,0	10,0	-316,7	-354,5
Total	68.654,6	1.829,0	32.420,0	1.110,0	52,8	39,3

Os resultados finais totais mostram que o espaço economizado variou de 45,38% (Tabela 5.1) a 39,3 - 52,8% (Tabela 5.2) o que se aproxima a 50%.

5.3.3 Análise de Tempo de Execução

Uma das preocupações da utilização de um SGBD no armazenamento dos dados do *pipeline* de um sequenciamento de alto desempenho é o custo em termos de tempo que será

necessário para a inserção dos dados das diferentes fases nas tabelas do esquema relacional. Por isso, foram realizadas algumas análises em relação a esse tempo de processamento.

As Tabelas 5.3 e 5.4 mostram os tempos gastos pelos processos de filtragem, mapeamento e análise comparado com o tempo gasto na inserção de dados no SGBD para os dados de célula de rim/figado e câncer de próstata. Na fase de filtragem foram armazenadas as SRS no SGBD junto com as SRS filtradas. Na fase de mapeamento as SRS foram mapeadas e os resultados (arquivos BAM) foram armazenados no SGBD. Na fase análise, a expressão diferencial foi realizada e os resultados armazenados no SGBD. Além disso, são mostrado os tempos de exportação dos dados inseridos no SGBD gerados nas diferentes fases.

Tabela 5.3: Comparação de tempo de procesamento e armazenamento (em SGBD) de dados de células de Rim/fígado.

	Processamento (hh:mm:ss)	**Inserção no SGBD (hh:mm:ss)**	**Exportação no SGBD (hh:mm:ss)**	**Tempo-Inserção (%)**	**Tempo-Exportação (%)**
Filtragem	01:51:22	01:51:54	00:28:27	100,4	25,5
Mapeamento	68:26:12	00:08:55	00:01:51	0,2	0,04
Análise	00:17:52	00:00:12	-	1,1	-
Total	70:35:26	02:01:01	00:30:18	2,9	0,7

Tabela 5.4: Comparação de tempo de procesamento e armazenamento (em SGBD) de dados de células de câncer de próstata LNCaP.

	Processamento (hh:mm:ss)	**Inserção no SGBD (hh:mm:ss)**	**Exportação no SGBD (hh:mm:ss)**	**Tempo-Inserção (%)**	**Tempo-Exportação (%)**
Filtragem	-	00:02:03	00:00:15	-	-
Mapeamento	05:10:35	00:00:14	00:00:02	0,08	0,01
Análise	00:15:50	00:00:13	-	1,4	-
Total	05:26:25	00:02:30	00:00:17	0,8	0,09

A Tabela 5.3 mosta que o tempo gasto pelos processos é maior em relação ao tempo gasto no armazenamento dentro do SGBD. O tempo de processamento dos dados de células de rim/fígado que mais demorou, foi do processo de mapeamento, 68 horas. Enquanto que o tempo de inserção e exportação no SGBD foi menos de 9 e 2 minutos respectivamente, o que representa apenas 0,2% e 0,04% em relação ao tempo do processo de mapeamento, respectivamente. Apenas na fase de filtragem, o tempo gasto para a execução do processo foi praticamente o mesmo para a inserção dos dados no SGBD. Este resultado é consequência dos mais de 64GB de dados (entre SRS e SRS filtradas) envolvidos no processo de filtragem fazendo que o tempo de inserção no SGBD seja ligeiramente maior (100,4%) em relação ao tempo gasto no processo de filtragem. Na fase de análise, o tempo gasto pelo processo de expressão diferencial é maior comparado com o tempo

de inserção dos dados no SGBD, os tempos de exportação na fase de análise não foram considerados neste estudo por não serem necessários. O tempo total de inserção e exportação no SGBD para os dados de células de rim e figado foi de 2,9% e 0.7% em relação ao tempo total gasto no *pipeline*. Este resultado mostra que a porcentagem do tempo de inserção e exportação são significativamente menos comparado ao tempo de total gasto pelo *pipeline*.

Os resultados de tempo gasto para os dados de câncer LNCaP mostrados na Tabela 5.4 são semelhantes aos obtidos com os dados de rim/fígado, onde o tempo gasto pelos processos é muito maior do que os tempos de inserção e exportação no SGBD. O tempo gasto no processo de filtragem foi nulo nesse estudo de caso, já que, as SRS dos dados de células de câncer LNCaP não possuírem as sequências de qualidades pelo qual não foi feito a filtragem das SRS. Já o tempo para inserção dessas SRS no SGBD foi de dois minutos e meio apenas. Como no caso anterior, o tempo total de inserção e exportação no SGBD foi de 0,8% e 0,09% em relação ao tempo total gasto no *pipeline* o que é bem menor em relação ao tempo de total gasto pelo *pipeline*.

5.4 Trabalhos Publicados

Durante o desenvolvimento da pesquisa apresentada nessa dissertação, foram produzidos alguns artigos, nos quais, um foi aceito como resumo estendido e dois foram aceitos como artigos completos como relatados a seguir.

O resumo estendido foi aceito e apresentado como pôster no XII *Brazilian Symposium on Bioinformatics*, sob o titulo de "*A Conceptual Model for Transcriptome High-Throughput Sequencing Pipeline*" [71]. O respectivo resumo estendido foi publicado nos *proceedings* do congresso pela Springer.

O artigo completo foi aceito no BIBM 2011, *Workshop on Data-mining of Next-Generation Sequencing Data*, sob o titulo de "*A Conceptual Data Model for Transcriptome Project Pipeline*" [72]. O respectivo artigo foi publicado nos anais do congresso.

O artigo completo foi aceito e apresentado no *The IADIS Applied Computing 2011 conference*, sob o titulo de "*A Data Base Schema for High-Throughput Sequencing Transcriptome Pipelines*" [73]. O respectivo artigo foi publicado nos anais do congresso.

Capítulo 6

Conclusões e Trabalhos Futuros

Nesta dissertação, foi realizado o estudo dos principais modelos de dados para a representação de dados biológicos disponíveis atualmente na literatura. A partir desse estudo, foi desenvolvido um modelo conceitual orientado a objetos para *pipelines* de sequenciamento de alto desempenho transcritômico baseado em três fases: filtragem, mapeamento e análise. A especificação do modelo proposto levou em consideração a necessidade dos projetos de sequenciamento envolvendo essas três fases, assim como também, suprir as deficiências apresentadas nos modelos da literatura.

O modelo conceitual desenvolvido nesta dissertação representa os dados gerados nas diferentes fases de um *pipeline* tais como SRS, SRS mapeadas, dados do genoma de referência e todos os processos envolvidos. Sendo assim, o modelo proposto contempla os dados biológicos e as informações sobre os processos envolvidos no *pipeline* de sequenciamento.

O esquema relacional foi baseado no modelo conceitual proposto. A especificação desse esquema relacional levou em consideração regras básicas para transformar um modelo conceitual em um esquema relacional, porém algumas dificuldades foram encontradas já que os dados dos sequenciamentos de alto desempenho tem características específicas, dentre essas destacam-se: a criação de tipo de dado para agrupar grandes quantidades de SRS e a criação de tabelas intermediárias entre o TranscriptDB gerado pelo GenomicFeatures e o subesquema da fase de análise; entre outras.

Após a implementação do esquema relacional foi avaliado o desempenho no armazenamento, levando em consideração o espaço economizado entre as abordagens SGBD e sistemas de arquivo. Os resultados obtidos nos dois estudos de caso demostraram que a abordagem SGBD em relação ao espaço economizado teve bons resultados de forma geral com 45,3% de espaço economizado para os dados do genoma de referência e TranscriptDB, de 39,3% para os dados de célula de rim/figado e 52,8% para os dados de células de câncer de próstata LNCaP. No primeiro estudo de caso (dados de células de rim e fígado), os dados são de volume consideráveis com o formato FASTQ gerado pelo sequenciador Illumina. No segundo estudo de caso (células de câncer de próstata LNCaP), os dados são menores em relação ao primeiro, mas o volume de dados é considerável.

Em relação ao tempo de processamento do *pipeline*, verificou-se que não é impactante a utilização de um SGBD nas fases de mapeamento e análise, uma vez que o tempo gasto para inserir e extrair os dados necessários para a execução do *pipeline* é pequeno em relação ao tempo total de processamento do mesmo. Já na fase de filtragem o tempo

gasto na inserção é ligeiramente maior ao tempo gasto da filtragem dos dados. Além disso, as atividades de entrada e saída de dados no SGBD durante a execução do *pipeline* é realizada com pouca frequência levando a que o tempo gasto por essas operações seja menor após terminado a análise do *pipeline* todo.

Os resultados obtidos demostram que a abordagem proposta ofereceu um grande avanço proporcionando melhoria na forma de armazenamento dos dados produzidos por um *pipeline* de sequenciamento de alto desempenho devido à economia de espaço, tempo baixo de inserção e exportação em relação ao tempo total gasto pelo *pipeline* todo e organização dos dados assim como todo beneficio que o uso de um SGBD traz. Além disso, o SGBD permite a implementação dos modelos de dados como o proposto neste trabalho, fornecendo as vantagem inerentes dos SGBD sobre o sistema de arquivos.

Em geral, acredita-se que o modelo proposto neste trabalho pode trazer muitas vantagens para uma abordagem SGBD assim como vantagens no desempenho para a gestão de grandes volumes de dados de sequenciamento de alto desempenho transcritômico.

Estudos futuros podem usar dados de diferentes tecnologias de sequenciamento, podendo identificar com mais precisão o nível de espaço economizado no armazenamento para dados mais diversificados. Consequentemente, investigar os efeitos que isto pode trazer no armazenamento dos dados por serem de diferentes fontes. Além disso, podem-ser feitas outros tipos de análises contemplando outras metricas além do espaço economizado e tempo gasto.

Uma decorrência natural deste trabalho é a implementação de novos processos na fase de análise do *pipeline* proposto. Foi adaptado um esquema de análise de expressão diferencial, no entanto, esquemas que armazenem dados para as análises filogenéticas, identificação de ncRNAs entre outras são desejáveis. Outra possibilidade é a integração ou expansão do modelo para lidar com dados procedentes de análises protéicas.

Um outro trabalho futuro é a integração de proveniência de dados. Recentemente, há um grande interesse na comunidade de banco de dados sob administração de proveniência de dados e é interessante observar como esta abordagem pode-se adaptar para dados de sequenciamento de alto desempenho transcritômico.

Referências

[1] P.A. Alvarez. Pipelines para transcritomas obtidos porsequenciadores de alto desempenho. Technical report, Departamento de Ciência da computação - Universidade de Brasília, 2009. x, 5, 10, 34

[2] T.C.C. da Silva. Som-portrait: um método para identificar rna não codificador utilizando mapas auto organizáveis. Technical report, Departamento de Ciência da computação - Universidade de Brasília, 2009. x, 5

[3] D.P. Alten. Estrutura quaternária de proteına. `http://www.daanvanalten.nl/quimica/module12/par01212protproducao.html`. Acessado em Dezembro, 2011. x, 6

[4] J.C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology*. PWS Publishing, January 1997. x, 4, 6, 7, 8, 9, 10, 14, 34

[5] N. W. Paton, S. A. Khan, A. Hayes, F. Moussouni, A. Brass, K. Eilbeck, C. A. Goble, S. J. Hubbard, and S. G. Oliver. Conceptual modelling of genomic information. *Bioinformatics*, 16(6):548–557, June 2000. x, 24, 25, 28, 29, 58

[6] R. Elmasri, F. Ji, J. Fu, Y. Zhang, and Z. Raja. Extending EER modeling concepts for biological data. In *Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems*, pages 599–604, Washington, DC, USA, 2006. IEEE Computer Society. x, 25, 26, 29, 58

[7] N. Busch and G. Wedemann. Modeling genomic data with type attributes, balancing stability and maintainability. *BMC Bioinformatics*, 10(1):97–113, 2009. x, 26, 27, 28, 29, 58

[8] J.A.F Macedo, F. Porto, S. Lifschitz, and P. Picouet. A conceptual data model language for the molecular biology domain. In *Proceedings of the Twentieth IEEE International Symposium on Computer-Based Medical Systems*, pages 231–236. IEEE Computer Society, 2007. x, 27, 28, 29, 58

[9] GUS2011DB, the genomics unified schema. `http://www.gusdb.org/about.php`. Acessado em Agosto, 2011. x, 29, 30

[10] C. J. Mungall, D. B. Emmert, and Consortium FlyBase. A Chado case study: an ontology-based modular schema for representing genome-associated biological information. 23(13):I337–I346+, 2007. x, 30, 31, 59

[11] M. Carlson, P. Aboyoun, H. Pagès, S. Falcon, and M Morgan. *Making and Utilizing TranscriptDb Objects*. BioConductor-Open source software for bioinformatics, Fevereiro 2012. xi, 54, 85

[12] A.M. Lesk. *Introduction to Bioinformatics*. Oxford University Press, May 2002. xii, 6, 7

[13] J. D. Watson and F. H. C. Crick. Molecular Structure of Nucleic Acids: A Structure for Deoxyribose Nucleic Acid. *Nature*, 171(4356):737–738, April 1953. 1, 8

[14] U. Röhm and J.A. Blakeley. Data management for high-throughput genomics. In *Conference on Innovative Data Systems Research (CIDR)*, volume 5667, pages 97–111, 2009. 1

[15] M.L. Metzker. Sequencing technologies - the next generation. *Nature reviews. Genetics*, 11(1):31–46, January 2010. 1

[16] S.A. Simon, J. Zhai, R.S. Nandety, K.P. McCormick, J. Zeng, and D.M. e Blake C. Mejia. Short-Read Sequencing Technologies for Transcriptional Analyses. *Annual Review of Plant Biology*, 60(1):305–333, January 2009. 1, 12, 32

[17] P. Clote and R. Backofen. *Computational Molecular Biology: An Introduction*. Wiley, 1 edition, September 2000. 6, 7, 8, 9

[18] J. Barciszewski and V.A. Erdmann. *Noncoding RNAs: molecular biology and molecular medicine*. Springer, 1 edition, January 2003. 9

[19] S. R. Eddy. Non-coding RNA genes and the modern RNA world. 2(12):919–929, 2001. 9

[20] D.W. Mount. *Bioinformatics: Sequence and Genome Analysis, Second Edition*. Cold Spring Harbor Laboratory Press, 2nd edition, July 2004. 12

[21] F. Sanger, S. Nicklen, and A.R. Coulson. Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences of the United States of America (PNAS)*, 74(12):5463–5467, 1977. 12, 13

[22] N. Hall. Advanced sequencing technologies and their wider impact in microbiology. *J Exp Biol*, 210(Pt 9):1518–1525, May 2007. 12

[23] E.R. Mardis. Next-generation dna sequencing methods. *Annual Review of Genomics and Human Genetics*, 9(1):387–402, 2008. 12

[24] H.V.F. Melo. Desenvolvimento de um pipeline para análise genômica e transcriptômica com base em web services. Master's thesis, Universidade Federal de São Carlos, 2010. 13, 33

[25] C. Baudet. Uma abordagem para trimagem, verificaçao de contaminaçao e clusterizaçao de seqüências est. Master's thesis, Universidade Estadual de Campinas (Unicamp), 2006. 14, 33

[26] M. Morgan, M. Carlson, V. Obenchain, D. Tenenbaum, and H. Pages. Genome news network. `http://www.genomenewsnetwork.org/resources/whats_a_genome/Chp3_1.shtml`. Acessado em Junho, 2011. 14

[27] C. Kanz, P. Aldebert, N. Althorpe, W. Baker, A. Baldwin, K. Bates, P. Browne, A. van den Broek, M. Castro, G. Cochrane, K. Duggan, R. Eberhardt, N. Faruque, J. Gamble, F.G. Diez, N. Harte, T. Kulikova, Q. Lin, V. Lombard, R. Lopez, R. Mancuso, M. Mchale, F. Nardone, V. S.ilventoinen, S. Sobhany, P. Stoehr, M.A. Tuli, K. Tzouvara, R. Vaughan, D. Wu, W. Zhu, and R. Apweiler. The EMBL Nucleotide Sequence Database. *Nucl. Acids Res.*, 33(suppl_1):D29–33, 2005. 15, 16

[28] D.A. Benson, I. KarschMizrachi, D.J. Lipman, J. Ostell, and D.L. Wheeler. GenBank: update. *Nucleic Acids Research*, 32(suppl 1):D23–D26, January 2004. 15

[29] H.M. Berman. The Protein Data Bank: a historical perspective. *Acta Crystallographica Section A*, 64(1):88–95, January 2008. 15

[30] R.L. Tatusov, N.D. Fedorova, J.D. Jackson, A.R. Jacobs, B. Kiryutin, E.V. Koonin, D.M. Krylov, R. Mazumder, S.L. Mekhedov, A.N. Nikolskaya, B.R. Rao, S. Smirnov, A.V. Sverdlov, S. Vasudevan, Y.I. Wolf, J.J. Yin, and D.A. Natale. The COG database: an updated version includes eukaryotes. *BMC bioinformatics*, 4(1):41–51, September 2003. 16

[31] H. Ogata, S. Goto, K. Sato, W. Fujibuchi, H. Bono, and M. Kanehisa. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic acids research*, 27(1):29–34, January 1999. 16

[32] R. Apweiler, T. K. Attwood, A. Bairoch, A. Bateman, E. Birney, M. Biswas, P. Bucher, L. Cerutti, F. Corpet, M. D. Croning, R. Durbin, L. Falquet, W. Fleischmann, J. Gouzy, H. Hermjakob, N. Hulo, I. Jonassen, D. Kahn, A. Kanapin, Y. Karavidopoulou, R. Lopez, B. Marx, N. J. Mulder, T. M. Oinn, M. Pagni, F. Servant, C. J. Sigrist, and E. M. Zdobnov. The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic acids research*, 29(1):37–40, January 2001. 16

[33] T. Pruitt, K. D. e Tatusova and D. R. Maglott. NCBI reference sequences (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*, 35(Database issue):D61–D65, January 2007. 16

[34] M. Worboys and M. Duckham. *GIS: A Computing Perspective, 2nd Edition.* CRC Press, Inc., Boca Raton, FL, USA, 2004. 17

[35] A. Silberschatz, H.F. Korth, and S. Sudarshan. *Database system concepts.* McGraw-Hill, New York, 6 edition, 2010. 17, 19, 34, 35, 55

[36] R. Elmasri and S. Navathe. *Fundamentals of Database Systems (6th Edition).* Addison Wesley, 6 edition, 2010. 18, 19, 22, 55

[37] C. Batini, S. Ceri, and S.B. Navathe. *Conceptual Database Design: An Entity-Relationship Approach.*, volume 116. Benjamin/Cummings, 1992. 18

[38] P.P. Chen. The entity-relationship modeltoward a unified view of data. *ACM Trans. Database Syst.*, 1:9–36, March 1976. 18

[39] M.P. Papazoglou, S. Spaccapietra, and Z. Tari, editors. *Advances in Object-Oriented Data Modeling*. MIT Press, Cambridge, MA, USA, 2000. 20

[40] J. Rumbaugh, M. Blaha, W. Premerlani, F. Eddy, and W. Lorensen. *Modelagem e projetos baseados em objetos*, volume 8. Campus, Rio de Janeiro, 1st edition, 1994. 20

[41] G. Booch, R.A. Maksimchuk, M.W. Engel, B.J. Young, J. Conallen, and K.A. Houston. *Object-Oriented Analysis and Design with Applications (3rd Edition)*. Addison-Wesley Professional, 3 edition, April 2007. 21

[42] M. Fowler. *UML Distilled: A Brief Guide to the Standard Object Modeling Language*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2003. 21

[43] G. Booch, J. Rumbaugh, and I. Jacobson. *UML - Guia Do Usuário*. Livraria Tempo Real Inform, 2 edition, 2005. 21

[44] E. F. Codd. A relational model of data for large shared data banks. *Commun. ACM*, 13(6):377–387, 1970. 22

[45] E.F. Codd. *Relational completeness of data base sublanguages*. IBM Corp., March 1972. 22

[46] E. Bornberg-Bauer and N.W. Paton. Conceptual data modelling for bioinformatics. *Brief Bioinform*, 3(2):166–180, January 2002. 24, 28, 29, 58

[47] D. Riehle, M. Tilman, and R. E. Johnson. Dynamic object model. In *2000 Conference on Pattern Languages of Programming (PLoP 2000)*, volume 5, pages 3–24, Washington University, Washington University, 2000. 26

[48] Chado. `http://gmod.org/wiki/Chado`. Acessado em Agosto, 2011. 29, 30

[49] C.V. Ibañez. Gus sb - a schema browser for the genomics unified schema (gus). Master's thesis, Graduate Faculty of The University of Georgia, 2009. 30, 59

[50] P. Zhou, D. Emmert, and P. Zhang. Using Chado to store genome annotation data. *Current protocols in bioinformatics / editoral board, Andreas D. Baxevanis ... [et al.]*, Chapter 9, January 2006. 30, 31

[51] Alicia Oshlack, Mark D. Robinson, and Matthew D. Young. From RNA-seq reads to differential expression results. *Genome biology*, 11(12):220–230, December 2010. 34

[52] L.D. Stein. Genome annotation: from sequence to biology. *Nature Reviews Genetics*, 2(7):493–503, July 2001. 34

[53] D. Frishman and Alfonso Valencia. *Modern Genome Annotation: The Biosapiens Network*. Springer Publishing Company, Incorporated, 1st edition, 2008. 34

[54] M. Morgan, M. Carlson, V. Obenchain, D. Tenenbaum, and H. Pages. High-throughput sequence analysis with r and bioconductor. http://www.bioconductor.org/help/course-materials/2011/SeattleIntro2011/Bioconductor-tutorial.pdf. Acessado em Julhio, 2011. 57

[55] J.C. Marioni, C.E. Mason, S.M. Mane, M. Stephens, and Y. Gilad. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research*, 18(9):1509–1517, September 2008. 52

[56] J.D. Hoheisel. Microarray technology: beyond transcript profiling and genotype analysis. *Nature reviews. Genetics*, 7(3):200–210, March 2006. 52

[57] H. Li, M.T. Lovci, Y.S. Kwon, M.G. Rosenfeld, X.D. Fu, and G.W. Yeo. Determination of tag density required for digital transcriptome analysis: application to an androgen-sensitive prostate cancer model. *Proceedings of the National Academy of Sciences*, 105(51):20179–20184, 2008. 52

[58] FastQC. http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/. Acessado em abril, 2011. 53

[59] G.J. Hannon. Fastx-toolkit. http://hannonlab.cshl.edu/fastx_toolkit/index.html. Acessado Abril, 2011. 53

[60] S. Andrews. Fastqc. a quality control tool for high throughput sequence data. http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/Help/1%20Introduction/. Acessado Abril, 2011. 53

[61] C. Trapnell, L. Pachter, and S.L. Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–1111, May 2009. 54

[62] B. Langmead, C. Trapnell, M. Pop, and S. Salzberg. Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biology*, 10(3):25–35, 2009. 54

[63] R Development Core Team. *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. 54

[64] R. Gentleman, V. Carey, D. Bates, Ben Bolstad, M. Dettling, Sandrine Dudoit, Byron Ellis, Laurent Gautier, Yongchao Ge, Jeff Gentry, Kurt Hornik, Torsten Hothorn, Wolfgang Huber, Stefano Iacus, Rafael Irizarry, Friedrich Leisch, Cheng Li, Martin Maechler, Anthony Rossini, Gunther Sawitzki, Colin Smith, Gordon Smyth, Luke Tierney, Jean Yang, and Jianhua Zhang. Bioconductor: open software development for computational biology and bioinformatics. *Genome Biology*, 5(10):R80+, 2004. 54

[65] N. Delhomme. *RNA-Seq Tutorial (EBI, October 2011)*, 2011. 54

[66] Mark D. Robinson, Davis J. McCarthy, and Gordon K. Smyth. edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, January 2010. 54

[67] S. Andrews. RdbiPgSQL. `http://rdbi.sourceforge.net`. Acessado Julio, 2011. 55

[68] N. Edelweiss. Bancos de dados temporais: teoria e prática. In *XVII Jornada de Atualização em Informática, do XVIII Congresso Nacional da Sociedade Brasileira de Computação*, volume 2, pages 225–282, 1998. 56

[69] David Salomon. *Data Compression : The Complete Reference*. Springer, February 2004. 56

[70] M. Morgan, M. Carlson, V. Obenchain, D. Tenenbaum, and H. Pages. Toast. `http://www.postgresql.org/docs/8.4/static/storage-toast.html`. Acessado em Julio, 2011. 60

[71] R.C. Huacarpuma, M. Holanda, and M.E.M.T. Walter. A conceptual model for transcriptome high-throughput sequencing pipeline. In *Proceedings of the 6th Brazilian conference on Advances in bioinformatics and computational biology*, BSB'11, pages 71–74, Berlin, Heidelberg, 2011. Springer-Verlag. 63

[72] R.C. Huacarpuma, M Holanda, and M.E.M.T. Walter. A conceptual data model for transcriptome project pipeline. In *BIBM Workshops*, pages 13–18. IEEE, 2011. 63

[73] R.C. Huacarpuma, M.T. Holanda, S. Lifschitz, and M.E.M.T. Walter. A database schema for high-throughput sequencing transcriptome pipelines. In *Proceedings of IADIS International Conference on Applied Computing*, pages 187–194, Novembro 2011. 63

Anexo I

Diagrama de Clases do Modelo Conceitual

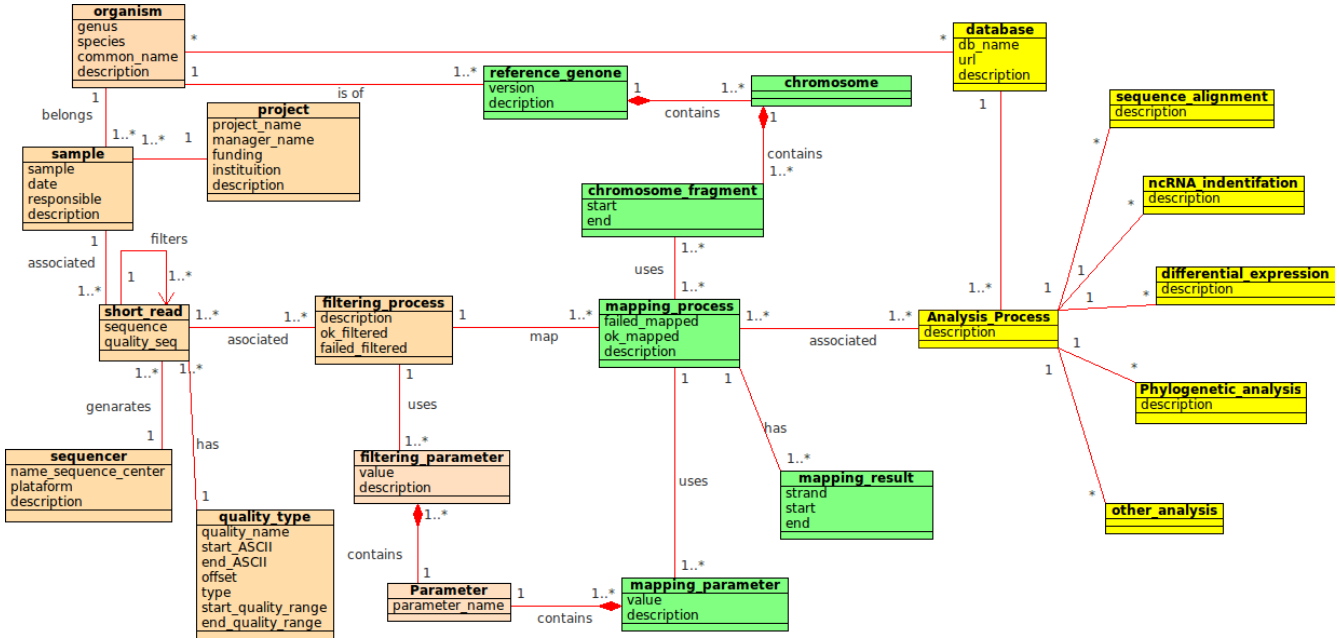

Figura I.1: Diagrama de classes do modelo conceitual para um *pipeline* de sequenciamento de alto desempenho transcritômico.

Anexo II

Esquema Relacional do *Pipeline*

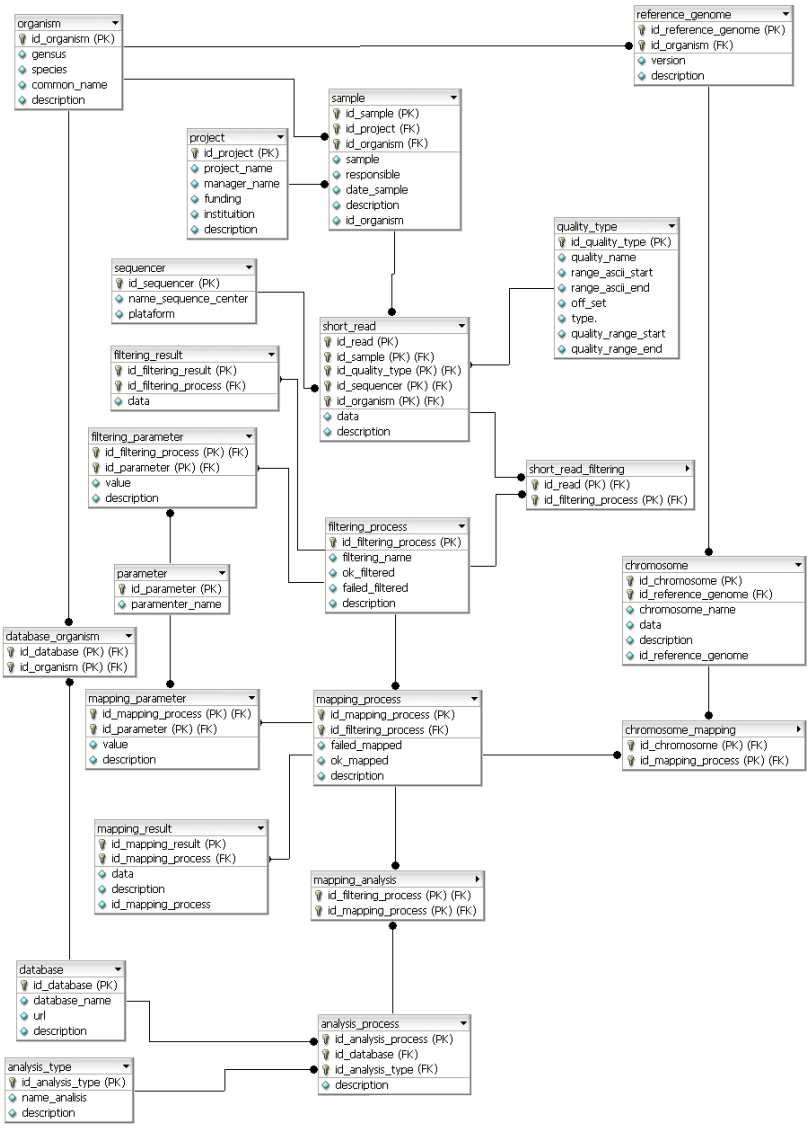

Figura II.1: Esquema relacional do *pipeline* de sequenciamento de alto desempenho transcritômico.As linhas ponteadas de cor cinza associam as tabelas *gene_result, transcript_result, cds_result e exon_result* com o esquema relacional de transcritos gerado pela ferramenta usada na fase de análise no estudo de caso.

Anexo III

Tabela do Esquema de Filtragem

Tabela III.1: Tabelas e colunas do subesquema filtragem.

Tabela	**Coluna**	**Descrição coluna**
organism	*id_ organism*	Identificador do organismo
	genus	Gênero do organismo
	species	Espécie do organismo
	common_ name	Nome comum do organismo
	description	Descrição do organismo
sample	*id_ sample*	Identificador da amostra
	sample	Nome da amostra
	date	Data de preparação da amostra
	responsible	Pessoa encarregada
	description	Descrição
project	*id_ project_ name*	Identificador do projeto
	project_ name	Nome do projeto
	namager_ name	Pessoa encarregada do projeto
	funding	Fundação que financia o projeto
	instituition	Instituição a cargo do projeto
	description	Descrição
short_ read	*id_ read*	Identificador do conjunto contendo as SRS
	id_ sample	Chave forânea da onde vem as SRS
	id_ quality_ type	Chave estrangeira da tabela *quality_ type*, indica o tipo de qualidade que das SRS
	id_ sequencer	Chave estrangeira da tabela *sequencer*, indica a tecnologia usada no sequenciamento
	data	Contém um conjunto de SRS
	description	descrição
		Continua na proxima pagina

Tabela III.1: (continuando)

Tabela	**Coluna**	**Descrição coluna**
sequencer	*id_ sequencer*	Identificador do sequenciador
	name_ sequencer_ center	Nome do centro de sequenciamento
	plataform	Tecnologia do sequenciador
	description	descrição
quality_ type	*id_ quality*	Identificador do tipo de qualidade
	quality_ name	Nome do tipo de qualidade
	start_ ASCII	Inicio do rango de caráteres ASCII
	end_ ASCII	Fim do rango de caráteres ASCII
	offset	Deslocamento
	type	Tipo de escore de qualidades (PHRED, Solexa, ...)
	start_ quality_ range	Inicio do rango do escore de qualidade
	end_ quality_ range	Fim do rango do escore de qualidade
filtering_ process	*id_ filtering_ process*	Identificador do processo de filtragem
	filtering_ name	Nome do processo de filtragem
	ok_ filtered	Número de SRS filtradas con sucesso
	failed_ filtered	Número de SRS filtradas sem sucesso
	description	Descrição do processo de filtragem
filtering_ paramenter	*id_ filtering_ process*	Chave forânea da tabela *filtering_ process*, indica o processo de filtragem
	id_ parameter	Chave forânea da tabela *parameter*, indica o parâmetro usado
	value	Valor do parâmetro
	description	Descrição do valor usado
parameter	*id_ parameter_ name*	Identificador do parâmetro
	parameter_ name	Nome do parâmetro
short_ read_ filtering	*id_ filtering_ process*	Chave forânea da tabela *filtering_ process*, indica o processo de filtragem
	id_ read	Chave forânea da tabela *short_ read*, indica um conjunto de SRS
		Continua na proxima pagina

Tabela III.1: (continuando)

Tabela	**Coluna**	**Descrição coluna**
filtering_result	*id_filtering_result*	Identificador do resultado da filtragem
	id_filtering_process	Chave forânea da tabela *filtering_process*, indica o processo de filtragem
	data	Contém um conjunto de SRS filtradas

Anexo IV

Tabela do Esquema de Mapeamento

Tabela IV.1: Tabelas e Colunas do subesquema mapeamento.

Tabela	**Coluna**	**Descrição coluna**
reference_ genome	*id_ reference_ genome*	Identificador do genoma de referência
	id_ organism	Chave estrangeira da tabela *organism*, identifica o organismos
	version	Versão do genoma
	description	descrição
chromosome	*id_ chromosome*	Identificador do cromossomo
	id_ reference_ genome	Chave estrangeira da tabela *reference_ genome*, identifica o genoma de referência.
	chromosome_ name	Nome do cromossomo
	data	Contém a sequência de bases do cromossomo
	description	descrição
chromosome_ mapping	*id_ mapping_ process*	Chave estrangeira da tabela *mapping_ process*, identifica o processo de mapeamento
	id_ chromosome	Chave estrangeira da tabela *chromosome*, associa o cromossomo com o processo de mapeamento
mapping_ process	*id_ mapping_ process*	Identificador do processo de mapeamento
	id_ filtering_ result	Chave estrangeira da tabela *filtering_ result*, identifica o resultado o resultado da filtragem
	failed_ mapped	Número de SRS não mapeadas
	ok_ mapped	Número de SRS mapeadas com sucesso
		Continua na proxima pagina

Tabela IV.1: (continuando)

Tabela	**Coluna**	**Descrição coluna**
	description	Descrição
mapping_parameter	*id_mapping_process*	Chave estrangeira da tabela *mapping_process*, identifica o processo de mapeamento
	id_parameter	Chave estrangeira da tabela *parameter*, associa o parâmetro usado
	value	Valor de parâmetro
	description	Descrição
mapping_resul	*id_mapping_result*	Identificador do resultado pro processo de mapeamento
	id_mapping_process	Chave estrangeira da tabela *mapping_process*, identifica o processo de mapeamento
	data	Contém as SRS mapeadas
	start	Começo de onde foi mapeado a SRS
	description	Descrição

Anexo V

Tabela do Esquema de Análise Usada nos Estudos Caso

Tabela V.1: Tabelas e Colunas do subesquema da análise - Expressão.

Tabela	**Coluna**	**Descrição coluna**
analysis_process	*id_analysis_process*	Identificador do processo de análise
	id_database	Chave estrangeira da tabela *database*, identifica o banco de dados usado
	id_analysis_type	Chave estrangeira da tabela *analysis_type*, identifica o tipo de análise
	description	descrição da análise
analysis_type	*id_analysis_type*	Identificador do tipo de análise
	name_analysis	Nome da análise
	description	descrição do tipo de análise
database	*db_name*	Nome do banco de dados usado
	url	Sitio web
	description	Descrição
database_organism	*id_database*	Chave estrangeira da tabela *database*, identifica o banco de dados
	id_organism	Chave estrangeira da tabela *organism*, identifica o organismo
gene	*gene_id*	Identificador do gene
	gene_name	Nome do transcrito
	_chrom_id	Chave estrangeira da tabela *chrominfo*, identifica o cromossomo
	strand	Sentido da cadeia (fita)
	start	Início do transcrito
	end	Fim do transcrito
		Continua na proxima pagina

Tabela V.1: (continuando)

Tabela	**Coluna**	**Descrição coluna**
transcript	*_tx_id*	Identificador do transcrito
	tx_name	Nome do transcrito
	_chrom_id	Chave estrangeira da tabela *chrominfo*, identifica o cromossomo
	strand	Sentido da cadeia (fita)
	start	Início do transcrito
	end	Fim do transcrito
genes_transcript	*gene_id*	Chave estrangeira da tabela *gene*, identifica o gene
	_tx_id	Chave estrangeira da tabela *trascript*, identifica o transcrito do gene
exon	*_exon_id*	Identificador do exon
	exon_name	Nome do exon
	_chrom_id	Chave estrangeira da tabela *chrominfo*, identifica o cromossomo
	strand	Sentido da cadeia (fita)
	start	Início do exon
	end	Fim do exon
transcript	*_cds_id*	Identificador do cds
	cds_name	Nome do cds
	_chrom_id	Chave estrangeira da tabela *chrominfo*, identifica o cromossomo
	strand	Sentido da cadeia (fita)
	start	Início do cds
	end	Fim do cds
splicing	*_tx_id*	Chave estrangeira da tabela *trascript*, identifica o transcrito do gene
	exon_rank	Posição do exon no *splicing*
	_exon_id	Chave estrangeira da tabela exon, identifica o exon
	_cds_id	Chave estrangeira da tabela *cds*, identifica o cds
gene_result	*id_analysis_process*	Chave estrangeira da tabela *analysis_process*, identifica o processo de análise
	id_mapping_process	Chave estrangeira da tabela *mapping_process*, identifica o processo de mappeamento
	gene_id	Chave estrangeira da tabela *gene*, identifica um gene
		Continua na proxima pagina

Tabela V.1: (continuando)

Tabela	**Coluna**	**Descrição coluna**
	count	Número de SRSs que foram mapeadas dentro do gene identificado por *gene_id*
transcript_result	*id_analysis_process*	Chave estrangeira da tabela *analysis_process*, identifica o processo de análise
	id_mapping_process	Chave estrangeira da tabela *mapping_process*, identifica o processo de mappeamento
	_tx_id	Chave estrangeira da tabela *transcript*, identifica um transcrito
	count	Número de SRSs que foram mapeadas dentro do transcrito identificado por *_tx_id*
exon_result	*id_analysis_process*	Chave estrangeira da tabela *analysis_process*, identifica o processo de análise
	id_mapping_process	Chave estrangeira da tabela *mapping_process*, identifica o processo de mappeamento
	_exon_id	Chave estrangeira da tabela *exon*, identifica um exon
	count	Número de SRSs que foram mapeadas dentro do exon identificado por *_exon_id*
cds_result	*id_analysis_process*	Chave estrangeira da tabela *analysis_process*, identifica o processo de análise
	id_mapping_process	Chave estrangeira da tabela *mapping_process*, identifica o processo de mappeamento
	_cds_id	Chave estrangeira da tabela *cds*, identifica um cds
	count	Número de SRSs que foram mapeadas dentro do cds identificado por *_cds_id*
chrominfo	*_chrom_id*	Identificador do cromossomo
	chrom	Nome do cromossomo
	length	Tamanho do cromossomo
	is_circular	Se o cromossomo é circular

Anexo VI

Esquema Relacional do *Pipeline* Usado nos Estudos de Caso

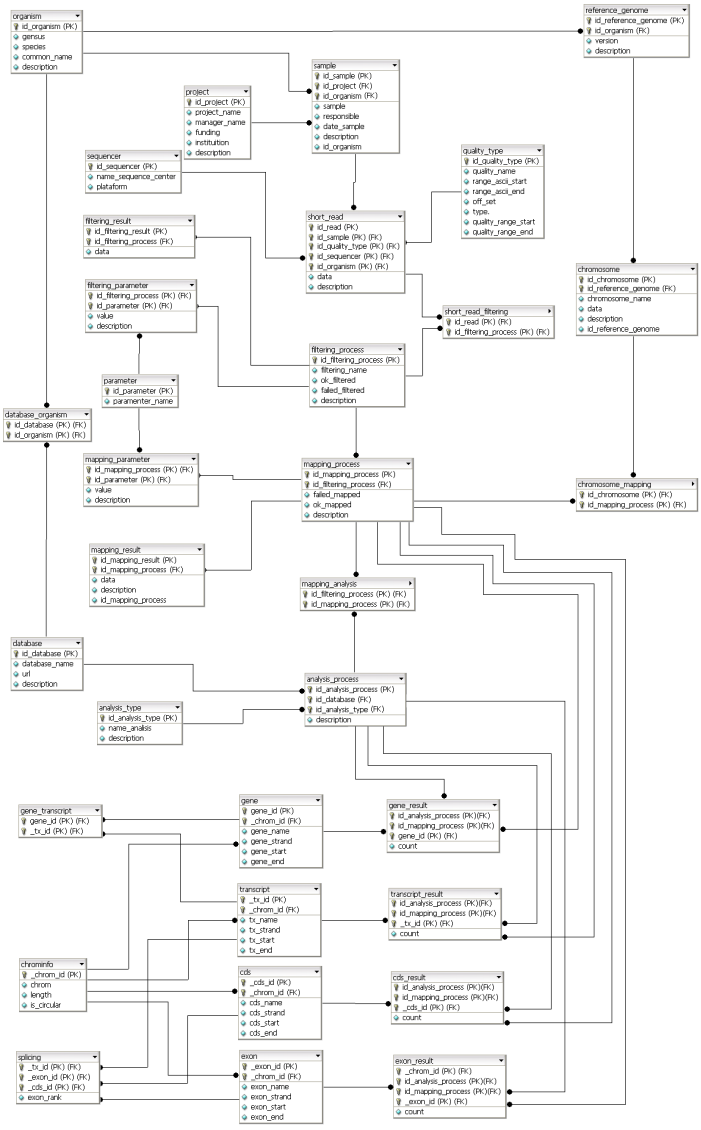

Figura VI.1: Esquema relacional do *pipeline* de sequenciamento de alto desempenho transcritômico. As linhas ponteadas de cor cinza delimita o esquema *TranscriptDB* gerado pelo pacote *GenomicFeatures* [11].

Anexo VII

Formato do Arquivo FASTQ

@SRR002325.1 080317_CM-KID-LIV-2-REPEAT_0003:2:1:906:788 length=36
GAGAACCCTTTCCTCTTAAATTCTACTTCCACATAA
+SRR002325.1 080317_CM-KID-LIV-2-REPEAT_0003:2:1:906:788 length=36
IIII:.GAIIIIII6III:%II=I;0I)>5*III3
@SRR002325.2 080317_CM-KID-LIV-2-REPEAT_0003:2:1:919:342 length=36
TGAACCTAGAGTCTGGATCTATTTTTGTCTGAATGC
+SRR002325.2 080317_CM-KID-LIV-2-REPEAT_0003:2:1:919:342 length=36
IIIIIIIII+IIIIIIFIII0IIIIIHIII)8)I5I
@SRR002325.3 080317_CM-KID-LIV-2-REPEAT_0003:2:1:874:773 length=36
GGTCGGTTCCTTCCTTTTTTGCCTAGATTTTATGTA
+SRR002325.3 080317_CM-KID-LIV-2-REPEAT_0003:2:1:874:773 length=36
IIIIIIIII+IIIIIIFIII0IIIIIHIII)8)I5I
@SRR002325.4 080317_CM-KID-LIV-2-REPEAT_0003:2:1:876:756 length=36
GGAAAGTTCTTACATCTTGCGACTCATGAAATATTT
+SRR002325.4 080317_CM-KID-LIV-2-REPEAT_0003:2:1:876:756 length=36
IIIIIIIII+IIIIIIFIII0IIIIIHIII)8)I5I
@SRR002325.5 080317_CM-KID-LIV-2-REPEAT_0003:2:1:893:816 length=36
GAAAGCGCTCAAGCTCAACACCCATCACCTAAAAAA
+SRR002325.5 080317_CM-KID-LIV-2-REPEAT_0003:2:1:893:816 length=36
IIIIIIIII+IIIIIIFIII0IIIIIHIII)8)I5I
@SRR002325.6 080317_CM-KID-LIV-2-REPEAT_0003:2:1:875:565 length=36
TGTTAATCTTCTGTCTTGTTTATCTTTGCAATATTG
+SRR002325.6 080317_CM-KID-LIV-2-REPEAT_0003:2:1:875:565 length=36
IIIIIIIII+IIIIIIFIII0IIIIIHIII)8)I5I