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

ncRNA-Agents: Anotação de RNAs não-codificadores baseado em Sistema Multiagentes

Wosley da Costa Arruda

Tese apresentada como requisito parcial
para conclusão do Doutorado em Informática

Orientadora
Prof.ª Dr.ª Maria Emilia M. T. Walter

Brasília
2014

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

Coordenador: Prof. Dr.ª Alba Cristina M. A. de Melo

Banca examinadora composta por:

Prof.ª Dr.ª Maria Emilia M. T. Walter (Orientadora) — CIC/UnB
Prof. Dr.ª Célia Ghedini Ralha — CIC/UnB
Prof.ª Dr.ª Membro 2 — CIC/UnB
Prof.ª Dr.ª Membro 3 — CIC/UnB
Prof.ª Dr.ª Membro 4 — CIC/UnB

CIP — Catalogação Internacional na Publicação

da Costa Arruda, Wosley.

ncRNA-Agents: Anotação de RNAs não-codificadores baseado em Sistema Multiagentes / Wosley da Costa Arruda. Brasília : UnB, 2014.
131 p. : il. ; 29,5 cm.

Tese (Doutorado) — Universidade de Brasília, Brasília, 2014.

1. RNAs não-codificadores, 2. Anotação de RNAs não-codificadores,
3. Sistemas Multiagentes, 4. Bioinformática, 5. Inteligência Artificial

CDU 10/0132871

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

ncRNA-Agents: Anotação de RNAs não-codificadores baseado em Sistema Multiagentes

Wosley da Costa Arruda

Tese apresentada como requisito parcial
para conclusão do Doutorado em Informática

Prof.ª Dr.ª Maria Emilia M. T. Walter (Orientadora)
CIC/UnB

Prof. Dr.ª Célia Ghedini Ralha
CIC/UnB

Prof.ª Dr.ª Membro 2
CIC/UnB

Prof.ª Dr.ª Membro 3
CIC/UnB

Prof.ª Dr.ª Membro 4
CIC/UnB

Prof. Dr.ª Alba Cristina M. A. de Melo
Coordenador do Doutorado em Informática

Brasília, 13 de Junho de 2014

Dedicatória

Ao meu pai, que plantou em mim a semente da curiosidade científica.

À minha mãe, pelo exemplo de forca e determinacao.

Devo tudo a vocês.

Agradecimentos

Resumo

Os RNAs não-codificadores (ncRNAs) constituem um importante subconjunto dos transcritos produzidos nas células dos organismos, pois afetam diversos processos celulares. Existem métodos computacionais bastante eficazes para identificar proteínas, mas a anotação de ncRNAs é hoje objeto de pesquisa intensa, pois suas características e sinais não são ainda completamente conhecidos. Neste contexto, este trabalho apresenta uma arquitetura para anotação de ncRNAs baseada no paradigma de Sistemas Multiagentes, e usa raciocínio baseado em regras declarativas para decidir a anotação de uma determinada sequência de RNA, como base nas predições de ferramentas conhecidas.

Palavras-chave: RNAs não-codificadores, Anotação de RNAs não-codificadores, Sistemas Multiagentes, Bioinformática, Inteligência Artificial

Abstract

The science...

Keywords: non-coding RNA, Annotation of Non-Coding RNAs, Multi-agent System, Bioinformatics, Artificial Intelligence.

Sumário

1 Introdução **1**
- 1.0.1 Motivação . . . 2
- 1.0.2 Problema . . . 2
- 1.0.3 Hipótese . . . 2
- 1.0.4 Objetivos . . . 2
- 1.0.5 Descrições dos Capítulos . . . 3

2 Biologia Molecular e Bioinformática **4**
- 2.0.6 Ácidos nucléicos . . . 4
- 2.0.7 Dogma Central da Biologia Molecular: Síntese de Proteínas . . . 8
- 2.0.8 *Pipelines* em projetos genoma . . . 12

3 RNAs não-codificadores **16**
- 3.0.9 Conceitos Básicos . . . 16
- 3.0.10 Classificações de ncRNAs . . . 17
- 3.0.11 Ferramentas computacionais para detecção de ncRNAs . . . 18
- 3.0.12 Bancos de dados . . . 21
- 3.0.13 Características e Desafios na Anotação de ncRNAs . . . 23

4 Sistema Multiagentes e Regras de Inferência **25**
- 4.0.14 Agentes . . . 25
- 4.0.15 SMAs . . . 26
- 4.0.16 Ferramentas para SMAs . . . 26
- 4.0.17 Motores de Regras de Inferência . . . 33

5 Projeto: ncRNA-Agents **36**
- 5.0.18 Arquitetura . . . 36
- 5.0.19 Detalhes de Implementação . . . 38

6 Resultados **44**
- 6.0.20 Obtidos . . . 45
- 6.0.21 Esperados . . . 46

7 Conclusão **47**
- 7.0.22 Disciplinas . . . 47
- 7.0.23 Disciplinas . . . 47
- 7.0.24 Pesquisa . . . 47

Referências **49**

Lista de Figuras

2.1 Estrutura esquemática de um nucleotídeo, mostrando seus principais componentes: açúcar, fosfato e base nitrogenada [9]. 4
2.2 Os quatros tipos de nucleotídeos que compõem a molécula de DNA. 5
2.3 Replicação DNA Semiconservativa. À esquerda podemos observar um trecho de uma molécula de DNA que evidencia o aspecto de dupla-hélice e à direita as fitas mãe separadas, servindo de molde para as filhas, resultando em duas moléculas idênticas à dupla-hélice original [51]. 6
2.4 Tipos de nucleotídeos que compõem a molécula de RNA, observando-se que a Uracila (U) substitui a Timina (T) nos RNAs. 6
2.5 Modelo da estrutura de um rRNA 16S da E.coli [31]. 8
2.6 A síntese protéica em célula procariótica funciona da seguinte maneira. No passo 1, fatores de transcrição ligam-se às regiões da regulação dos genes que controlam, ativando-os. No passo 2, a RNA polimerase começa a transcrição do gene ativado pela região promotora, resultando na formação do mRNA. No passo 3, o mRNA e ligado no ribossomos. Nesse ponto, a proteína é sintetizada pelo ribossomo que liga os aminoácidos em uma cadeia linear [51]. 10
2.7 A síntese protéica em célula eucariótica funciona da seguinte maneira. No passo 1, fatores de transcrição ligam-se às regiões da regulação dos genes que controlam, ativando-os. No passo 2, a RNA polimerase começa a transcrição do gene ativado pela região promotora, resultando na formação do pré-mRNA. No passo 3, é feito um processamento na transcrição para remover sequências não-codificadoras. Para finalizar, no passo 4, o mRNA move-se para o citoplasma e é ligado pelo ribossomos. Nesse ponto, a proteína é sintetizada pelo ribossomo que liga os aminoácidos em uma cadeia linear [51]. 11
2.8 *pipeline* Sanger. 13
2.9 Um *pipeline* genérico para o sequenciamento do 454 Roche. 15
2.10 Um *pipeline* genérico para o sequenciamento do Illumina. 15

4.1 Arquitetura de um agente inteligente (Adaptado de [79]). 25
4.2 Estrutura de um Sistema Multi-Agente (Adaptado de [102]). 26

5.1 Arquitetura do ncRNAs-Agents. [4]. 37
5.2 Página do ncRNA-Agents, mostrando as ferramentas e os bancos de dados usados nos experimentos. 40

5.3 *Sniffer* dos agentes do ncRNA-Agents: a camada de resolução de conflitos aguardando a resposta do Agente Gerente Homologia com a ferramenta Infernal. 41
5.4 *Sniffer* dos agentes do ncRNA-Agents: a camada de resolução de conflitos enviando resposta para a interface do ncRNA-Agents para o usuário. . . . 42

6.1 Página do ncRNA-Agents, mostrando as ferramentas e os bancos de dados usados nos experimentos. 44
6.2 *Sniffer* dos agentes do ncRNA-Agents: Aguardando tomada de decisão da camada RC. 45
6.3 *Sniffer* dos agentes do ncRNA-Agents: Enviando resposta para a interface. 45

Lista de Tabelas

2.1 O Código Genético [97]. 12

3.1 Alguns tipos de ncRNAs e suas funções conhecidas [19, 86] 18
3.2 Ferramentas computacionais 21
3.3 Bancos de Dados 22

4.1 Avaliação das Ferramentas 32
4.2 Avaliação dos Motores de Inferência. 35

5.1 ncRNAs identificados no Projeto Genoma Pb. 43
5.2 ncRNAs identificados no Projeto Genoma Guaraná. 43

6.1 ncRNAs identificados no Projeto Genoma Pb. 46
6.2 ncRNAs identificados no Projeto Genoma Guaraná. 46

Capítulo 1

Introdução

Desde o trabalho de [101], em que eles propuseram a estrutura para uma molécula de DNA, a comunidade científica vem realizando um grande esforço para tentar compreender a estrutura e o funcionamento da biologia molecular nos seres vivos. Na década de 1990, iniciou-se um consórcio internacional, que teve o intuito de mapear e sequenciar o genoma humano por completo. Concluído em 2001, esse projeto sequenciou o genoma humano com 3 bilhões de bases e cerca de 20.000 a 30.000 genes [45, 86, 95].

A Bioinformática utiliza conhecimentos das áreas da Computação, Matemática e Estatística, com a finalidade de resolver problemas de Biologia Molecular. Nesta área, são desenvolvidos *pipelines* e ferramentas computacionais para apoiar os biólogos em projetos de sequenciamento de genomas de modo a converter dados experimentais em informações biologicamente relevantes [73, 82, 83]. Nesses projetos, o enorme volume de dados gerados aliado à complexidade dos problemas de Biologia Molecular requerem técnicas sofisticadas de computação, e constituem hoje objetos de pesquisa importantes na área própria de Computação.

Até a década de 1990, as moléculas de ácidos ribonucléicos (RNAs) estavam relacionadas ao uso da informação contidas no DNA para a tradução de proteínas, como o RNA mensageiro (mRNA), com exceção apenas do RNA transportador (tRNA) e do RNA ribossomal (rRNA), que também desempenham funções relacionadas diretamente à tradução de proteínas [86]. Porém, desde então, descobriram-se outros tipos de moléculas de RNA, que não são traduzidas em proteínas e estão presentes nos organismos afetando uma grande variedade de processos celulares. Essas moléculas, antes chamadas de lixo, são hoje denominadas de RNAs não-codificadores (ncRNAs) [49].

Os ncRNAs controlam uma gama notável de reações biológicas e processos, como iniciação da tradução, controle da abundância mRNA, arquitetura do cromossomo, manutenção de células-tronco, desenvolvimento do cérebro, músculos e secreção de insulina, dentre outras [61].

Apesar de sua importância funcional, e de muitas pesquisas buscarem classicar e identificar ncRNAs, os métodos biológicos e computacionais ainda não são capazes de identificá-los e classificá-los, o que afeta diretamente a anotação de ncRNAs.

Do ponto de vista experimental, os ncRNAs são caracterizados pela ausência de tradução em proteínas. Do ponto de vista computacional, o fato de sequências de certas classes de ncRNAs serem curtas e não terem um padrão de sequência bem comportado impedem que sejam reconhecidos apenas pelas suas bases (sequências primárias), o que

significa que em geral devem ser caracterizadas pelas suas estruturas secundárias [39]. Uma observação importante é a de que, em geral ncRNAs não podem ser identificados e classificados pelas mesmas ferramentas que detectam genes codificadores de proteína de forma tão eficiente, como o BLAST [76].

Por outro lado, Sistemas Multiagentes (SMAs), dentro de Inteligência Artificial, caracterizam-se pela distribuição da inteligência entre diferentes entidades autônomas (agentes), que interagem para atingir objetivos individuais ou coletivos. Para tanto, os agentes que compõem um SMA precisam negociar, cooperar para atingir objetivos (que não podem ser realizados por um só agente) e coordenar esforços conjuntos [102].

Este trabalho propõe o uso de um SMA para auxiliar na anotação de ncRNAs, particularmente utilizando ferramentas baseadas em técnicas de raciocínio automatizado e aprendizagem de máquina.

1.0.1 Motivação

Identificar e anotar ncRNAs constituem-se hoje em pesquisas desafiadoras tanto em Biologia Molecular, quanto em Bioinformática, devido a descobertas recentes de que ncRNAs exercem funções diversificadas e importantes nos mecanismos celulares, como a regulação do metabolismo de outras moléculas, o auxílio do transporte de proteína, a edição de nucleotídeos, a regulação de *imprinting* e estado da cromatina [85]. A tendência atual de pesquisa é usar várias ferramentas diferentes e os biólogos usarem seu raciocínio biológico para decidir a anotação de sequências que potencialmente seriam ncRNs.

1.0.2 Problema

Tanto quanto sabemos, não há ferramenta computacional usando simulação do raciocínio dos biólogos para, a partir do resultado de diversas ferramentas, recomendar anotação de ncRNAs.

1.0.3 Hipótese

Uma abordagem baseada em SMA será eficaz para criar uma ferramenta de anotação de ncRNAs, pois poderá combinar várias metodologias usando regras de inferência, usadas para simular o raciocínio dos biólogos.

1.0.4 Objetivos

Principal

O objetivo deste trabalho é propor um SMA para a anotação de ncRNAs, denominado ncRNA-Agents, utilizando diversas ferramentas e bancos de dados, conhecidos e regras de inferência para simular o raciocínio dos biólogos.

Específicos

- Propor uma arquitetura baseada em SMA para anotação de ncRNAs;

- Implementar uma ferramenta baseada na arquitetura do item anterior, e disponibilizar pela web;
- Realizar testes com dados reais de projetos de sequenciamento de genomas;
- Comparar a ferramenta com outras existentes na literatura.

1.0.5 Descrições dos Capítulos

No Capítulo 2, serão abordados conceitos relativos à Biologia Molecular e Bioinformática.

No Capítulo 3, serão ncRNAs. Serão apresentadas algumas classes conhecidas de ncRNAs, e elencados desafios para detecção e anotação de ncRNAs, que motivaram o desenvolvimento desta pesquisa. Neste mesmo capítulo, também são apresentados repositórios (base de dados) e ferramentas comumente usadas para detectar ncRNAs.

No Capítulo 4, são apresentados SMAs, particularmente conceitos básicos e propriedades. Serão também exploradas ferramentas oara implementar SMAs.

No Capítulo 5, apresentamos uma arquitetura do ncRNA-Agents, um SMA para anotar ncRNAS. Será detalhado o protótipo implementado, as ferramentas e bancos de dados usados, além dos resultados obtidos e dos resultados almejados quando esta tese for concluída.

Por fim, no Capítulo **??**, apresentaremos as atividades já realizadas, um cronograma com as atividades futuras, além das contribuições esperadas.

Capítulo 2

Biologia Molecular e Bioinformática

Neste capítulo, serão apresentados conceitos básicos de Biologia Molecular e Bioinformática, necessários ao entendimento deste trabalho. Na Seção 2.0.6, são apresentados os ácidos nucléicos, que contêm informações e mecanismos para sintetizar proteínas. Na Seção 2.0.7, descrevemos o Dogma Central da Biologia Molecular, ou o processo através do qual as informações contidas no DNA, através de diversos tipos de RNAs, são utilizados para a síntese de proteínas. Na Seção 2.0.8 particularmente descreve mas técnicas de sequenciamento de genomas e os *pipelines* associados.

2.0.6 Ácidos nucléicos

Os ácidos nucléicos são polímeros formados a partir de moléculas mais simples, chamadas de nucleotídeos. Um nucleotídeo tem na sua composição açúcar com cinco átomos de carbono (pentose), ligado a um grupo fosfato e uma base nitrogenada [42], como podemos ver na Figura 2.1.

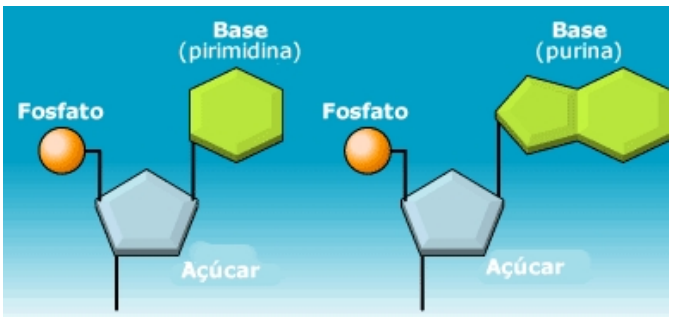

Figura 2.1: Estrutura esquemática de um nucleotídeo, mostrando seus principais componentes: açúcar, fosfato e base nitrogenada [9].

A síntese dos ácidos nucléicos envolve ligações entre diferentes nucleotídeos, atráves dos grupos fosfato, por meio de uma ligação chamada ligação fosfodiéster. Nessa ligação, o átomo de fósforo estabelece fortes ligações covalentes com os átomos de carbono da pentose dos nucleotídeos.

DNA

O ácido desoxirribonucléico (DNA) contém as informações genéticas de um organismo vivo, com exceção de alguns vírus [87]. Nos organismos procariotos, o material genético está espalhado na célula. Nas células eucarióticas, o DNA está localizado no núcleo, e contém informações sobre como, quando e onde produzir cada tipo de proteína [51].

Uma molécula de DNA é formada por cadeias de nucleotídeos. As bases nitrogenadas, as quais constituem os nucleotídeos, podem ser divididas em dois grupos: purinas ou pirimidinas. Nas bases purinas encontram-se as bases formadas por dois anéis, a citosina (C) e a timina (T) e nas pirimidinas encontram-se as bases formas por anéis, a Adenina (A) e a Guanina (G). Assim, existem quatro tipos de nucleotídeos, de acordo com sua base nitrogenada Figura 2.2, onde P representa o grupo fosfato, D a pentose (açúcar denominado de desoxirribose) as bases nitrogenadas que podem ser A, G, C ou T.

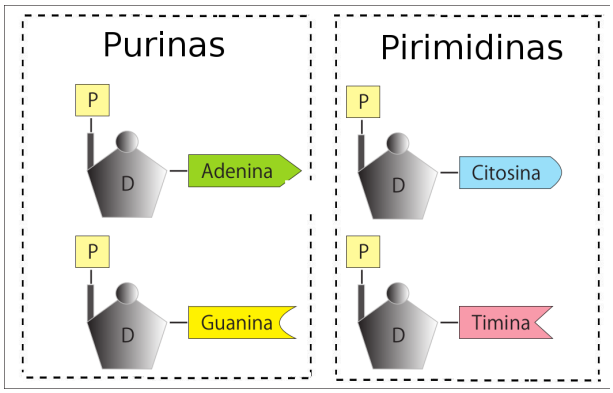

Figura 2.2: Os quatros tipos de nucleotídeos que compõem a molécula de DNA.

A estrutura do DNA é formada por duas cadeias ou fitas paralelas compostas por uma sequência de nucleotídeos, que são unidas através de pontes de hidrogênio formadas entre as bases nitrogenadas de cada fita, onde a base A estará pareada com a base T (A-T) e C com G (C-G) [92]. As bases A-T e C-G são chamadas de bases complementares. Essas duas fitas estão dispostas em espiral em torno de um eixo. Cada fita possui uma extremidade chamada 5' e a outra chamada 3', o que cria uma orientação em cada uma das fitas. As duas cadeias ficam, portanto em direção antiparalelas (opostas), formando uma dupla-hélice Figura 2.3.

O DNA pode sofrer replicações em alguns momentos através de um processo conhecido como duplicação semiconservativa, onde cada DNA recém formado possui uma das cadeias da molécula mãe [52]. Para realizar a replicação, a dupla fita do DNA abre-se, através do rompimento das pontes de hidrogênio, e os nucleotídeos livres encaixam-se na molécula através de novas pontes de hidrogênio. Os nucleotídeos vão sendo ligados entre si pela enzima DNA polimerase. O resultado desse processo é a formação de duas moléculas de DNA idênticas à original Figura 2.3.

Algumas regiões do DNA possuem informações para codificar proteínas ou ncRNAs e são chamadas de genes. Esses genes são transcritos em RNAs, que realizam diversas funções, como catalíticas ou estruturais para síntese de proteínas ou regulação. Assim, a sequência de nucleotídeos do DNA é chamada de estrutura primária, ou seja, a estrutura

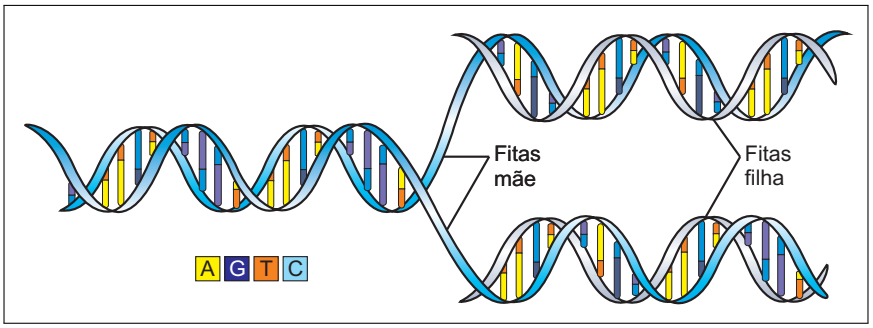

Figura 2.3: Replicação DNA Semiconservativa. À esquerda podemos observar um trecho de uma molécula de DNA que evidencia o aspecto de dupla-hélice e à direita as fitas mãe separadas, servindo de molde para as filhas, resultando em duas moléculas idênticas à dupla-hélice original [51].

primária é dada pela sequência linear formada ao longo da cadeia, sendo o nível estrutural mais simples [51]. A estrutura secundária consiste no arranjo espacial da sequência.

RNA

De forma semelhante ao DNA, o ácido ribonucléico (RNA) é uma molécula constituída por cadeias de nucleotídeos, ou polinucleotídeo.

O RNA é também formado pelo grupo fosfato, açúcar (ribose) e por uma base nitrogenada. Porém, entre as bases nitrogenadas, a Uracila (U) substitui a Timina (T) do DNA [51]. Assim, U é uma base purina. Na Figura 2.4, P representa o grupo fosfato, R a ribose e, em seguida, as bases nitrogenadas A, G, C e U.

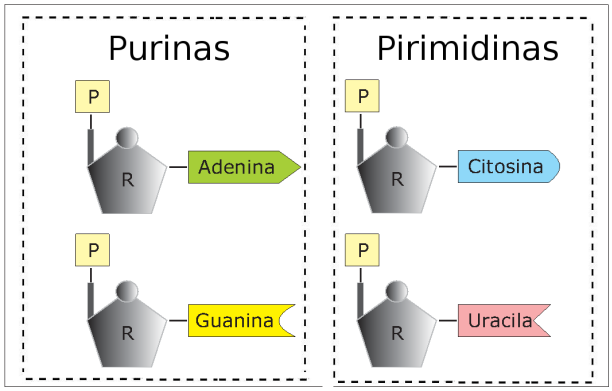

Figura 2.4: Tipos de nucleotídeos que compõem a molécula de RNA, observando-se que a Uracila (U) substitui a Timina (T) nos RNAs.

O RNA é formado em geral por uma única fita de nucleotídeos, e não possui o aspecto de dupla-hélice [42]. Porém, como o RNA tem como bases complementareas (A-U) e (C-G), pode unir-se através de pontes de hidrogênio, ou seja, o RNA pode dobrar-se.

Na síntese de proteínas estão envolvidos três tipos de RNAs, descritos com mais detalhes na Seção 2.0.7: o mensageiro (mRNA), o ribossômico (rRNA) e o transportador (tRNA).

A Figura 2.5 mostra uma estrutura de um rRNA 16S da bactéria *Escherichia coli*. Esse RNA é o componente central dos ribossomos, que são organelas encontradas no citoplasma possuindo duas subunidades chamadas de 40S e 60S em células eucarióticas e 30S e 50S em bactérias [43]. O mRNA é encontrado tanto no núcleo (onde ocorre sua síntese) quanto no citoplasma (onde participa da tradução de proteínas). Por último, vamos falar sobre os tRNAs, que são encontrados no citoplasma e funcionam durante a tradução, fazendo as ligações entre as proteínas e ácidos nucléicos. Esses RNAs são pequenos, contendo entre 70 e 90 nucleotídeos [42].

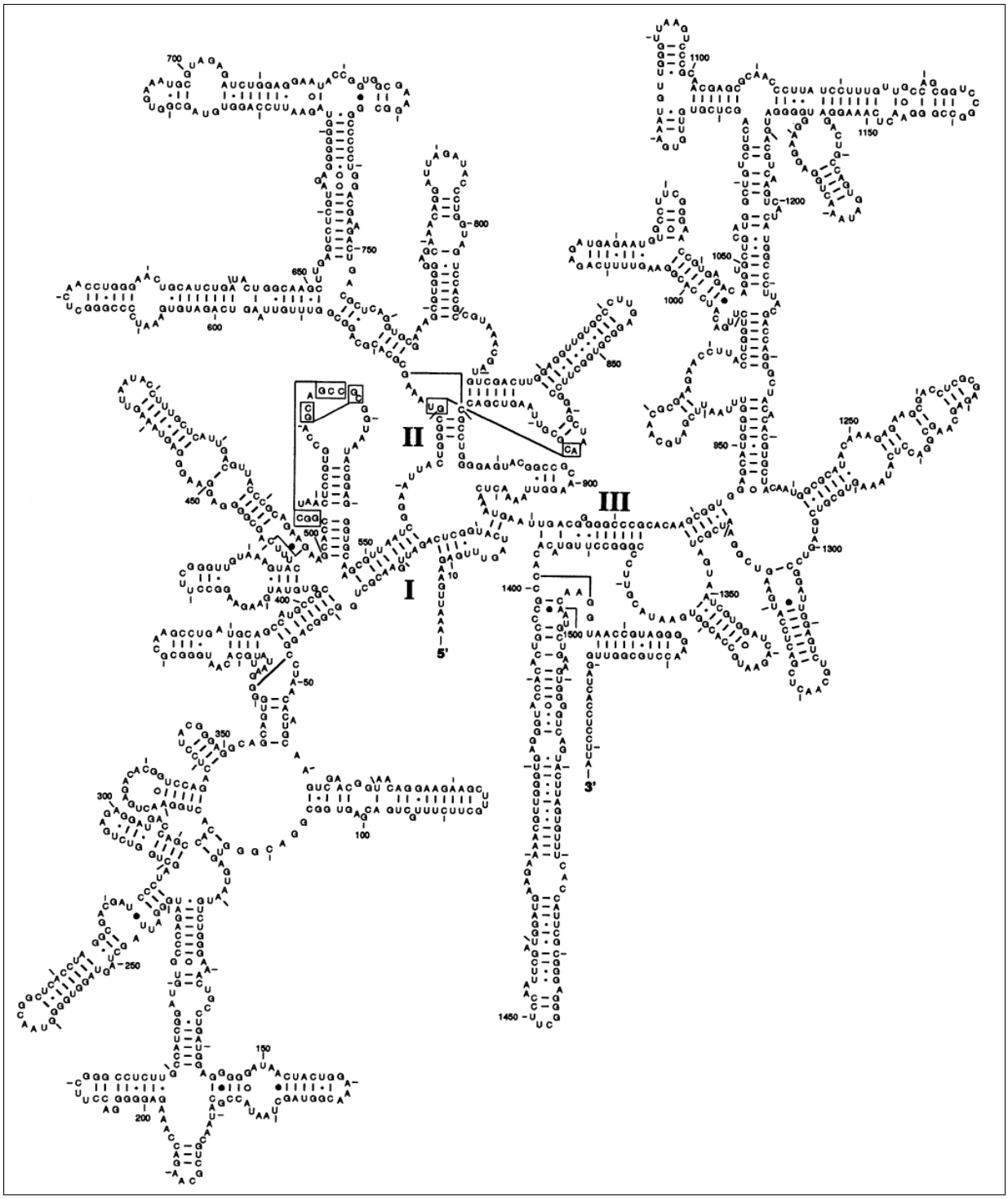

Figura 2.5: Modelo da estrutura de um rRNA 16S da E.coli [31].

2.0.7 Dogma Central da Biologia Molecular: Síntese de Proteínas

Nesta seção, mostraremos como as informações presentes em uma molécula de DNA são utilizadas na síntese de proteína.

Em primeiro lugar, a transcrição ocorre no núcleo, e é sintetizada pela enzima RNA polimerase, que vai ligar-se a uma determinada sequência de nucleotídeos do DNA, iden-

tificadas pelas regiões promotoras, e a percorre utilizando-a como molde até encontrar as regiões terminadoras. A transcrição baseia-se no pareamento de bases complementares usando uma fita do DNA como molde, ou seja, adenina com uracila (A $\rightarrow$ U), timina com adenina (T $\rightarrow$ A) e citosina com guanina (C $\leftrightarrow$ G). A molécula de RNA recém sintetizada é o mRNA [51].

O processo de transcrição acontece tanto nos procariotos (não possuem núcleo celular) Figura 2.6 como nos eucariotos (tem o DNA armazenado em um núcleo celular), tendo seu processo de transcrição um pouco mais complexo que os procariotos.

De acordo com [51], todos os organismos possuem maneiras de controlar quando os seus genes podem ser transcritos. Muitas células são capazes de responder a sinais externos ou a alterações nas condições externas, ligando ou desligando genes específicos, dessa forma, as células adaptam-se às necessidades do momento.

Nos eucariotos, o mRNA recém-trancrito é conhecido como pré-mRNA, e em alguns organismos ele irá sofrer algumas modificações antes que se transforme em um mRNA maduro [87]. No decorrer desse processo de maturação ocorre o *splicing* (eliminação dos íntrons do pré-mRNA) Figura 2.7, que são regiões que não codificam a proteína [51, 54].

O mRNA formado através da transcrição move-se para o citoplasma, precisamente nos ribossomos, onde ocorre o segundo processo para a síntese da proteína: a tradução.

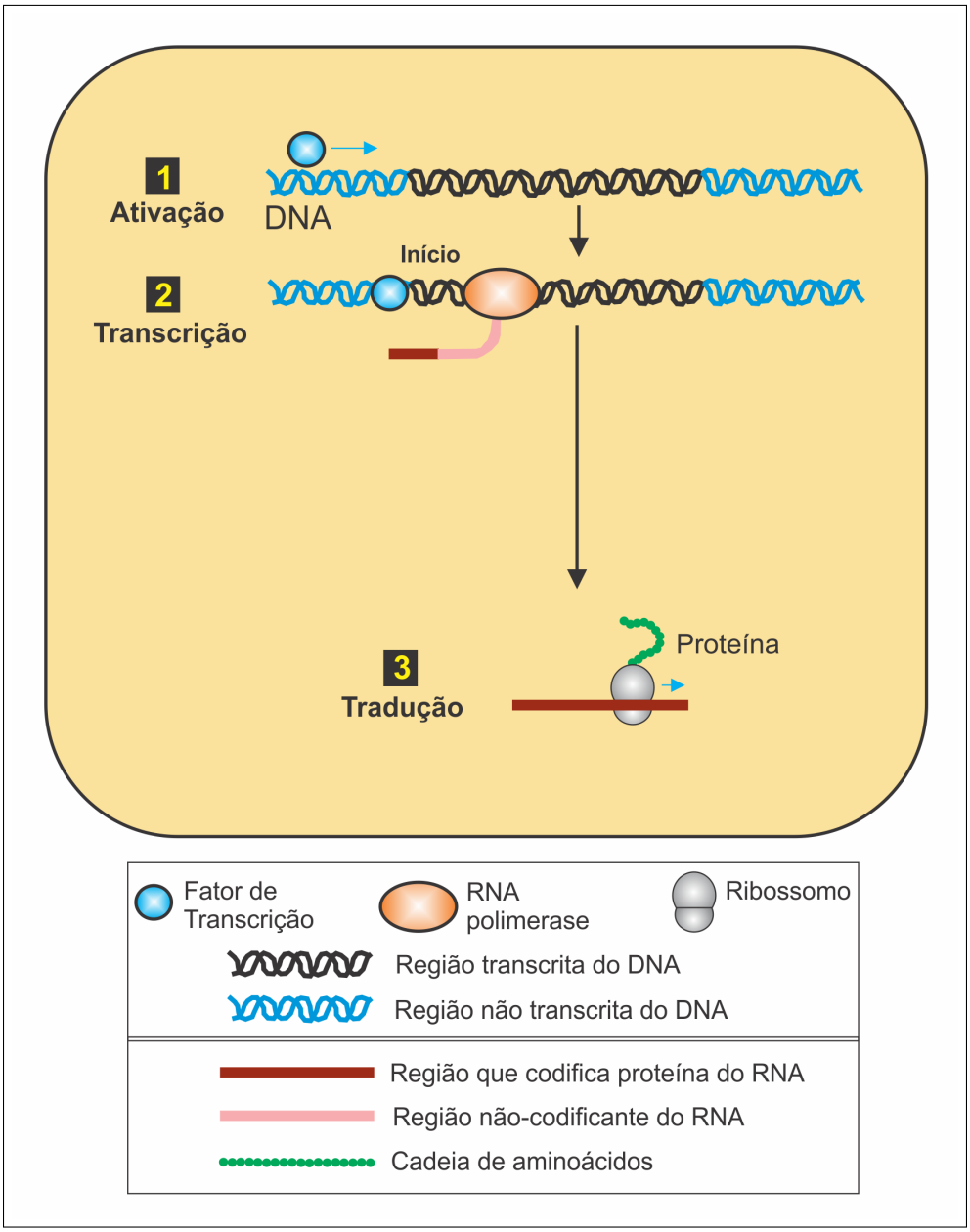

Figura 2.6: A síntese protéica em célula procariótica funciona da seguinte maneira. No passo 1, fatores de transcrição ligam-se às regiões da regulação dos genes que controlam, ativando-os. No passo 2, a RNA polimerase começa a transcrição do gene ativado pela região promotora, resultando na formação do mRNA. No passo 3, o mRNA e ligado no ribossomos. Nesse ponto, a proteína é sintetizada pelo ribossomo que liga os aminoácidos em uma cadeia linear [51].

No processo de tradução, primeiramente o mRNA liga-se entre as duas subunidades do ribossomo, onde cada códon do mRNA é pareado com o anticódon correspondente que está presente em moléculas de tRNA [87]. Cada aminoácido é codificado por um grupo de três bases do DNA, recebendo o nome de *tríplex* ou *códon*. Cada códon corresponde a um único aminoácido, porém um mesmo aminoácido pode ser definido por mais de um códon Tabela 2.1 [52, 87]. Existem ainda três códons (UAG, UAA e UGA) que não

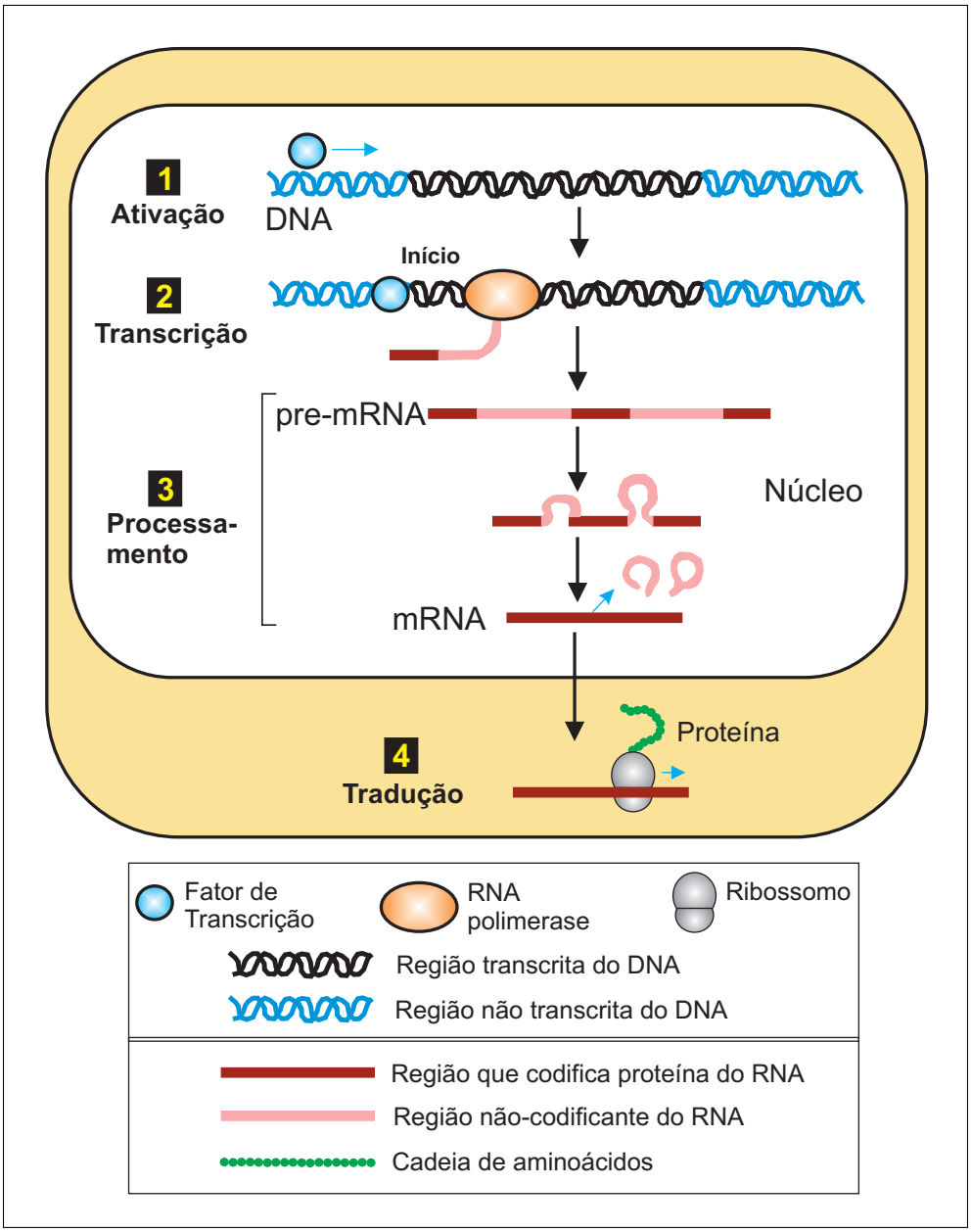

Figura 2.7: A síntese protéica em célula eucariótica funciona da seguinte maneira. No passo 1, fatores de transcrição ligam-se às regiões da regulação dos genes que controlam, ativando-os. No passo 2, a RNA polimerase começa a transcrição do gene ativado pela região promotora, resultando na formação do pré-mRNA. No passo 3, é feito um processamento na transcrição para remover sequências não-codificadoras. Para finalizar, no passo 4, o mRNA move-se para o citoplasma e é ligado pelo ribossomos. Nesse ponto, a proteína é sintetizada pelo ribossomo que liga os aminoácidos em uma cadeia linear [51].

correspondem a nenhum aminoácido, mas idicam sinais de término da tradução [87].

Depois, é feito o pareamento do segundo tRNA, os aminoácidos são ligados e o primeiro tRNA é liberado. Esse processo é repetido até que apareça um sinal de terminação no mRNA, que vai resultar na formação de uma cadeia polipeptídica.

Tabela 2.1: O Código Genético [97].

Primeira posição	Primeira posição				Terceira posição
	U	**C**	**A**	**G**	
U	UUU Phe	UCU Ser	UAU Tyr	UGU Cys	**U**
	UUC Phe	UCC Ser	UAC Tyr	UGC Cys	**C**
	UUA Leu	UCA Ser	UAA Stop	UGA Stop	**A**
	UUG Leu	UCG Ser	UAG Stop	UGG Trp	**G**
C	CUU Leu	CCU Pro	CAU His	CGU Arg	**U**
	CUC Leu	CCC Pro	CAC His	CGC Arg	**C**
	CUA Leu	CCA Pro	CAA Gln	CGA Arg	**A**
	CUG Leu	CCG Pro	CAG Gln	CGG Arg	**G**
A	AUU Ile	ACU Thr	AAU Asn	AGU Ser	**U**
	AUC Ile	ACC Thr	AAC Asn	AGC Ser	**C**
	AUA Ile	ACA Thr	AAA Lys	AGA Arg	**A**
	AUG Met[a]	ACG Thr	AAG Lys	AGG Arg	**G**
G	GUU Val	GCU Ala	GAU Asp	GGU Gly	**U**
	GUC Val	GCC Ala	GAC Asp	GGC Gly	**C**
	GUA Val	GCA Ala	GAA Glu	GGA Gly	**A**
	GUG Val	GCG Ala	GAG Glu	GGG Gly	**G**

[a] AUG faz parte do sinal de inicialização.

As proteínas são compostos orgânicos constituídos por aminoácidos unidos através de ligações peptídicas. Elas estão envolvidas em todos os processos biológicos dos seres vivos, como nas funções estruturais catalizadoras e reguladoras [50].

2.0.8 *Pipelines* em projetos genoma

Em projetos de genomas, o sistema computacional é chamado *pipeline* [47], e é desenvolvido no laboratório de bioinformática. Descrevemos nesta seção, os *pipelines* da antiga tecnologia Sanger e as modernas, chamadas de sequenciamento de nova geração.

Já nas atuais tecnologias de processamento de alto desempenho, o *pipeline* é dividido em outras fases, e fases semelhantes com objetivos diferentes. A primeira fase o "mapeamento", busca alinhar os Expressed Sequence Tags (ESTs) obtidos durante um transcritoma com os de um genoma de referência, às vezes, o genoma do próprio organismo de onde os ESTs derivaram. No entanto, outros organismos podem ser utilizados. A fase

de anotação apresenta certas diferenças tendo como principal objetivo a identificação de RNAs não codificadores (ncRNA).

Técnica: *Sanger*

Na antiga tecnologia de sequenciamento Sanger, o *pipeline* é, em geral, dividido em três fases, **submissão**, **montagem** e **anotação**. A submissão visa receber as sequências geradas por sequenciadores automáticos a partir de experimentos realizados nos laboratórios de Biologia Molecular, transformando-as em cadeias de caracteres e armazenando-as em bancos de dados. Nota-se que estas sequências são fragmentos de DNA ou de RNA, pois a tecnologia sanger não consegue sequenciar o DNA inteiro nem mesmo mesmo uma molécula inteira de RNA.

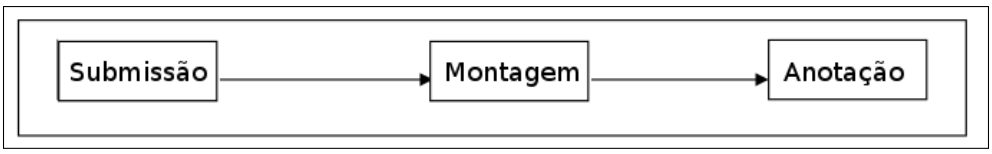

Figura 2.8: *pipeline* Sanger.

Durante a fase de montagem, as sequências que provavelmente vieram da mesma região do DNA devem ser agrupadas. Cada grupo constituído de duas ou mais sequências é denominado de *contig*, tendo cada *contig* uma sequência resultante (consenso) do agrupamento de todas as sequências que o formam. As sequências que não puderam ser agrupadas são denominadas de *singlets*.

A última fase, denominada de anotação, consiste em associar funções biológicas às sequências resultantes da fase de montagem (consensos das *contig* e *singlets*), tomando funções conhecidas de sequências similares que estão disponibilizadas em bancos de dados biológicos públicos. Essa fase é dividida em duas etapas: automática e manual.

A anotação automática compara as sequências geradas no projeto com sequências de bancos de dados privados e/ou públicos [8]. A comparação de duas sequências visa encontrar quais partes das sequências são parecidas ou similares. Dizemos que duas sequências são similares quando são "aproximadamente iguais", ou seja, têm exatamente os mesmos caracteres, com poucos caracteres diferentes, neste caso, podemos inferir que as duas sequências exercem o mesmo "papel biológico"ou têm a mesma função. Existem alguns métodos de comparação aproximada de sequências como BLAST [1] e FASTA [70], bastante utilizados para inferir funções para as sequências identificadas em um projeto genoma.

Na anotação manual, são utilizadas as informações da anotação automática, bem como o conhecimento do biólogo, para determinar a função que vai ser associada efetivamente à sequência analisada.

Na **submissão** de um projeto genoma feito em sequenciadores Sanger, depois da recepção dos arquivos com os resultados do sequenciamento de cada fragmento (também denominado *read*), um programa chamado *Phred* [20], traduzir as informações recebidas em sequências de letras contendo as bases identificadas e as probabilidades de erros associadas a cada base, gerando dois arquivos com extensão (*.phd*), para cada *read*. Depois, o programa *Phd2Fasta* cria, pelo *Phred*, dois arquivos texto no formato FASTA: um con-

tendo a sequência de bases nitrogenadas e outro contendo os valores das probabilidades de erro.

A fase de submissao contém um grande numero de sequências, porém, nem todas são utilizadas nos próximos passos. Para ser utilizada, uma sequência deve possuir uma probabilidade de erro suficientemente baixa, sendo essa probabilidade determinada de acordo com cada projeto. Serão aceitas as sequências com qualidade mínima, o que aumenta a confiabilidade nos dados.

O sequenciamento Sanger envolve cópias do DNA do organismo estudado, inseridas em um outro organismo. Assim, as sequências podem não pertencer ao organismo que está sendo estudado. Um programa chamado *Cross match* visa identificar e retirar vetores e contaminantes das sequências.

Na fase de **montagem**, são usado em geral dois programas, o *CAP3* [37] ou o *Phrap* [26]. Estes programas geram arquivos FASTA contendo: as sequências de todos os *singlets*; dados sobre a composição e sequências dos *contigs*; e informações gerais sobre a montagem dos fragmentos de DNA.

Existe dois tipos de projetos de sequenciamento: genômico e transcritômico. No primeiro caso, para a identificação dos possíveis genes presentes nos *contigs* e *singlets* obtidos durante a montagem, utiliza-se o programa *Glimmer* [81], para identificar ORFs. No caso de transcritomas, os *contigs* e *singlets* já codificam um gene, e assim, não há necessidade de se utilizar o *Glimmer* para fazer a idenficação.

Por fim, na fase de **anotação** é utilizado o programa BLAST para fazer a comparação das sequências identificadas com bancos de dados de sequências cujas funções já são conhecidas na fase de anotação automática. Depois de feito esse procedimento, os bíologos podem proceder a anotação manual das sequências de acordo com seus conhecimentos.

Em todas etapas anteriormente descritas, são armazenadas estatísticas sobre o projeto, sendo algumas delas: o número de sequências aceitas e rejeitadas na fase de submissão, número de *contigs* e *singlets* encontrados (montagem), além de anotações manuais e automáticas.

Técnica: Alto-Desempenho

No início da década de 2000, novas técnicas de sequenciamento massivamente pararelas revolucionaram a forma de realizar o sequenciamento do DNA. Com um custo muito baixo em comparação com o método Sanger, permitindo o sequenciamento de milhões de sequências, esses métodos tiveram um grande impacto nas áreas de pesquisa onde se realiza sequenciamento do DNA, e assim abriram novas frentes de pesquisas, como o estudo de DNAs antigos em mamutes, e diversidade ecológica por meio do sequenciamento de DNA de amostras ambientais [56].

Nesta seção estudaremos o funcionamento de dois sequenciadores, o 454-FLX da Roche e o Illumina

454-FLX Roche

O 454-FLX foi o primeiro sequenciador a aparecer no mercado, em 2004, utilizando uma técnica de sequenciamento conhecida com o nome de pirosequenciamento[78]. Essa técnica busca incorporar os nucleotídeo a uma fita de DNA por meio da enzima DNA polimerase que acarreta a liberação de pirofostato. Com isso a molécula inicia uma série

de reações químicas cujo produto final é a liberação de luz. A detecção da luz e obtida por um sensor que permite a determinação das bases de uma sequência de DNA.

O 454-FLX Roche gera sequências de cerca de 250 a 600 bases de comprimento. Depois desse processamento, sequências com baixa qualidade são removidas, obtendo em cerca de milhões de bases com boa qualidade em média, com cerca de 1 milhão de *reads*. O tamanho das sequências obtidas com o sequenciamento 454-FLX é menor que o sequenciamento Sanger, mas foi utilizado com sucesso no sequenciamento de genomas virais de bactériais com uma qualidade bastante alta.

Em geral, *pipelines* para o 454 Roche envolvem uma fase de **filtragem**, que retira sequências com qualidades ruins, uma fase de **montagem**, feita por sobreposição de extremidades similares de sequências de entrada, e uma fase de **análise**, que inclui anotação.

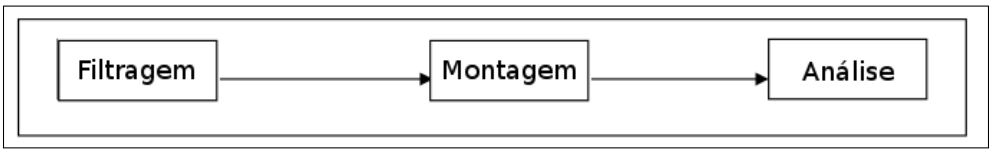

Figura 2.9: Um *pipeline* genérico para o sequenciamento do 454 Roche.

Illumina

A técnica de sequenciamento utilizada no Illumina incorpora um nucleotídeo por vez à sequência que está sendo determinada. Primeiro, realiza-se a amplificação de fragmentos do DNA, sendo incorporadoes adaptadores no início e fim de cada um dos fragmentos de DNA, que são anexandos a uma superfície. Depois disso, a DNA polimerase é utilizada para a produção de grupos de sequências, cada grupo contendo aproximadamente 1 milhão de cópias de fragmentos de DNA original.

Depois de amplificado o DNA, realiza-se o processo de sequenciamento em si. Nesse estágio, nucleotídeos fosforecentes são adicionados às moléculas de DNA amplificadas. Com a incorporação concluída, processa-se a imagem com as luzes oriundas dos nucleotídeos fosforecentes. Esse processo continua por um determinado número de ciclos, sendo controlado pelo operador do sequenciador, e podem ser construídas as sequências com 25 a 90 bases de comprimento, o que gera bilhões de bases sequenciadas, e cerca de 10 milhões de *reads* [14].

Um *pipeline* para Illumina inclui uma fase de **mapeamento**, onde as *reads* de tamanho curto são mapeadas em um genoma de referência, e uma fase de **análise**, que inclui estatísticas de cobertura de **análise** de expressão diferencial, por exemplo.

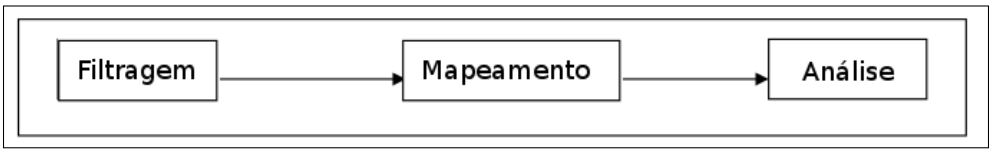

Figura 2.10: Um *pipeline* genérico para o sequenciamento do Illumina.

Capítulo 3

RNAs não-codificadores

Neste capítulo descreveremos os RNAs não-codificadores (ncRNAs), que serão objeto de estudo deste trabalho de doutorado. Na seção 3.0.9, estudamos ncRNAs de forma geral. Na Seção 3.0.10, são apresentadas classificações e descrições dos ncRNAs. Em seguida, na Seção 3.0.11, são descritas algumas ferramentas computacionais para detecção de ncRNAs das classes conhecidas. Na Seção 3.0.12, mostraremos os bancos de dados utilizados. Na Seção 3.0.13, seguem características e desafios na detecção de ncRNAs, que serviram de motivaram para esta pesquisa.

3.0.9 Conceitos Básicos

RNA não-codificadores (ncRNA) são tipos específicos de RNA que não codificam proteínas [19, 101], entretanto sua conformação permite que desempenhem funções em vários processos celulares.

O primeiro enunciado do Dogma Central da Biologia Molecular dizia que a função dos RNAs, restringia-se a participação em síntese de proteínas. Contudo, estudos recentes revelam que a quantidade de ncRNAs equivale a 98% do genoma humano. Uma quantidade tão grande de ncRNAs não pode simplesmente estar sendo produzida sem razão. Portanto atualmente vários estudos pretendem compreender melhor o papel desses RNAs nos organismos.

Apesar de identificados, e possuírem papel de grande importância, a caracterização dos RNAs não-codificadores foi, nas décadas de 1980 e 1990, abandonada e relegada para um segundo plano, talvez devido a dificuldades técnicas relacionadas a identificar essas moléculas pequenas, instáveis e pouco abundantes [19]. Nesta época, os RNAs não envolvidos diretamente com a síntese de proteínas eram chamados de DNA lixo (*junk* DNA) [86].

A partir do início dos anos 2000, retomaram-se os estudos dos ncRNAs, em consequência da crescente quantidade de ncRNAs identificados pelos biólogos e descritos na literatura. As descobertas mais notáveis envolvendo RNAs estruturais estão relacionadas ao desenvolvimento do sistema nervoso, corroborando a observação de que a quantidade de regiões não-codificadoras é proporcional à complexidade dos organismos [6, 57, 60, 72]. Além dos *small interfering* RNA (siRNA) e micro RNAs (miRNA) que ocorrem nos organismos, foram desenvolvidos novos métodos artificiais baseados nos mecanismos nos

quais esses RNAs estavam envolvidos, citando principalmente silenciamento de RNAs mensageiros alvo, que já são empregados com sucesso [74].

Os métodos computacionais para identificar ncRNAs sofrem de problemas similares aos dos métodos experimentais. A Bioinformática não possui métodos únicos para identificação e classificação de ncRNAs, embora alguns critérios sejam usados, como: o fato de que ncRNAs não possuem em geral ORFs longas; há ao longo de sua sequência ocorrências de códons de parada maiores do que a esperada [98]; provavelmente RNAs tem uma conservação em nível de estrutura secundária e não primária, o que invalida a detecção de ncRNAs por meio de ferramentas tradicionais usadas para caracterizar similaridade de DNA em proteínas [66, 76, 93]. Estudos que incorporam uso de códons, substituições sinônimas e não-sinônimas e energia mínima de dobramento também são bem-sucedidos na identificação de ncRNAs [6, 93, 104].

Geralmente, os ncRNAs não possuem uma sequência conservada, tendo como principal característica a conservação de sua estrutura espacial que inclui bidimensional ou tridimensional, tornando sua identificação mais difícil. Os ncRNAs mais conhecidos possuem uma estrutura tridimensional complexa e têm funções tanto catalisadoras como estruturais [16]. A atual tendência dos métodos de Bioinformática é recorrer a uma combinação de diversos métodos computacionais que caracterizem os ncRNAs por meio de diferentes princípios, e depois analisar todas as informações geradas pelos métodos para decidir quais RNAs provavelmente são não-codificadores [50, 62].

Sabendo que o problema na busca de ncRNA é essencialmente de classificação, a escolha de um método depende dos dados disponíveis das sequências que estão sendo analisadas [76, 99]. Uma desvantagem é que para confirmar a predição feita *in silico* de um transcrito computacional ser codificador, que exige validação experimental no laboratório, a observação de ausência de tradução não é conclusiva, já que o eventual transcrito poderia ser traduzido logo que exposto a outras condições, ambientais ou fisiológicas [50, 62].

Desde a descoberta dos ncRNAs, muitas perguntas têm sido feitas e muitos estudos têm sido direcionados à procura de respostas. Porém, essas moléculas ainda não são bem conhecidas. A principal causa disso é que maior parte das pesquisas para detecção de genes, durante muito tempo, foram voltadas na direção de RNAs mensageiros e proteínas.

3.0.10 Classificações de ncRNAs

Os ncRNAs, são classificados, em geral, de acordo com suas funcionalidades, o que permite entender diferentes aspectos relacionados às suas sequências genômicas e, portanto, ao papel que os ncRNAs realizam nos mecanismos celulares. Na Tabela 3.1 são tratadas classes de ncRNAs importantes.

O **RNA transportador (tRNA)** [69] é utilizado como molécula transportadora de informação de cada códon componente do mRNA em um aminoácido específico a ser adicionado à proteína sendo formada. O tRNA desempenha essa função através de duas regiões, o anticódon que é responsável pelo reconhecimento de códons específicos do mRNA, e o aminoácido correspondente ao códon.

O **RNA ribossomal (rRNA)** [19] é o componente central do ribossomo. A função do rRNA é prover um mecanismo para decodificar o mRNA em aminoácidos e interagir com os tRNAs durante a tradução, de síntese de proteína.

Tabela 3.1: Alguns tipos de ncRNAs e suas funções conhecidas [19, 86]

Classes de ncRNAs		
Sigla	**Função**	**Referências**
tRNA	envolvidos na tradução de mRNAs	[69]
rRNA	RNA constituinte do ribossomo	[19]
snoRNA	envolvidos na modificação do rRNA	[15]
miRNA	família putativa de genes reguladores da tradução.	[59]
siRNA	moléculas ativas na interferência de RNA	[19]
piRNA	regulação de tradução e estabilidade de mRNA, entre outras	[10]
snRNA	incluem RNAs relacionados ao processo de excisão	[92]
snmRNA	essencialmente pequenos RNAs não-codificadores	[38]
stRNA	interrompem a tradução de mRNA	[19]
rasiRNA	silenciamento da transcrição via remodelagem da cromatina	[103]

O ***small nucleolar*** **RNA (snoRNA)** [15] é uma classe de pequenas moléculas que realizam modificações químicas no rRNA, além de outros ncRNAs, tal como o tRNA. Essas modificações possuem como principal objetivo promover a maturação desses ncRNAs, transformando-os em moléculas ativas. A origem desses genes ainda não está clara, mas acredita-se que eles originam-se dos íntrons do mRNA.

O **microRNA (miRNA)** [59] parece estar relacionado com a regulação gênica. As moléculas de miRNA são parcialmente complementares a uma ou mais moléculas de mRNA e sua principal função é reduzir a expressão de genes codificadores, inibindo a tradução de mRNAs.

small interfering **RNA(siRNA)** [19] possui o mesmo papel do miRNA, porém reduz a expressão de genes codificadores degradando o mRNA em vez de inibir sua tradução.

piwi-interacting **RNA (piRNA)** [10] é uma classe de pequenas moléculas de RNA existentes basicamente nas células de mamíferos. Assim como os miRNAs e os snoRNAs, os piRNAs também estão relacionados com a regulação gênica. Mais especificamente, eles atuam no silenciamento de genes capazes de se auto-duplicar no interior do genoma.

Small nuclear **RNA (snRNA)** [92] é uma classe de RNAs não-codificantes encontrados no interior do núcleo das células. O núcleo contém muitos tipos de snRNAs. A estrutura secundária desses RNAs é altamente conservada nos organismos. Alguns deles, conhecidos como U1, U2, U4, U5 e U6, são essenciais para o *splicing* do pre-mRNA.

small non-messenger **RNA (snmRNA)** [38] é uma classe de ncRNAs, com função reguladora.

Small temporal **RNA (stRNA)** [19] é uma classe de RNAs com função de regulação do desenvolvimento biológico.

repeat-associated **siRNA (rasiRNAs)** [103] estão envolvidos na manutenção da metilação do DNA e histonas em retrotranposons, são sequências repetitivas como as codificadoras de RNA ribossomal. A formação de heterocromatina, que ao nível do DNA é composta de sequências de retrotransposons degenerados e arranjos em tandem de unidades de repetições simples, também é mediada por rasiRNAs.

3.0.11 Ferramentas computacionais para detecção de ncRNAs

Nesta seção, descrevemos ferramentas para identificar e classificar de ncRNAs.

BLAST

O programa *Basic Local Alignment Search Tool* (BLAST) [1] é bastante utilizado na comparação entre sequências. É um método de alinhamento local, que compara uma sequência com outras sequências com funções já definidas em um banco de dados. O BLAST pode ser utilizado para identificar relação evolucionária, inferir funções ou identificar uma família de genes entre sequências.

O BLAST é constituído por uma série de programas:

- **blastp**: para comparação de sequências de aminoácidos com um banco de dados de proteínas;
- **blastn**: para comparação de sequências de nucleotídeos com banco de dados de DNA;
- **blastx**: para comparação de sequências de nucleotídeos traduzida em todas as ORFs com banco de dados de proteínas;
- **tblastn**: para comparação de sequências de proteínas com um banco de dados de sequências de nucleotídeos traduzidas em todas as suas ORFs;
- **tblastx**: para comparar as ORFs de sequências de nucleotídeos com todas as ORFs de um banco de dados de nucleotídeos.

Infernal

O Infernal (*INFERence of RNA Alignment*) é um método que utiliza uma abordagem baseada em Gramática Estocástica Livres de Contextos (SCFG, *Stochastic Context-Free Grammars*) [18, 80]. Essa ferramenta constrói perfis de RNA consenso chamados de modelos de covariância (CM, *Covariance Models*), que é um caso especial de SCFG projetado para modelar sequências e estruturas de RNAs. O Infernal usa esses CMs para procurar as semelhanças entre as estruturas secundárias das famílias de RNAs do banco de dados Rfam e da sequência com qual está sendo investigada.

Cada CM é construído a partir do alinhamento múltiplo de sequências e de dados relacionados à estrutura secundária consenso, em posições onde o alinhamento é único e onde ocorrem pareamento de bases. Pontuações são atribuídas para cada posição específica assim como para quantidades de nucleotídeos, pareamento de bases, inserções e deleções.

O Infernal compreende os programas *cmbuild* , *cmsearch* e *cmalign*. A construção do CM, feita pelo *cmbuild*, requer como entrada um alinhamento múltiplo de RNAs no formato Estocolmo (*Stockholm*) [17], gerando um arquivo de saída contendo o modelo de covariância, o qual será usado por outras funções do Infernal.

A busca em bases de dados por possíveis homólogos, feitas pelo *cmsearch* requer duas entradas: um arquivo CM (obtido com o *cmbuild*); e um arquivo contendo as sequências a serem analisadas.

O *cmsearch* busca as sequências que geraram *hits* com alta pontuação para o modelo de convariância usado. É gerada uma saída contendo os alinhamentos para cada *hit* em um formato similar a estrutura BLAST [1]. Já o alinhamento de possíveis homólogos, utilizando o *cmalign* requer um arquivo de CM e outro arquivo que contenha possíveis homólogos. Este programa alinha sequências de acordo com o CM, criando um alinhamento

múltiplo no formato Estocolmo. Esse alinhamento poderá ser utilizado como entrada na construção de um modelo de covariância pelo *cmbuild*, como descrito acima.

Dentro da ferramenta Infernal, existe um módulo chamado *Rsearch*, que compara sequências de RNAs com um banco de sequências conhecidas de RNAs. Desta forma, dada uma sequência de RNA, são feitas buscas em uma base de dados de nucleotídios por RNAs homólogos. Esta busca é baseada tanto na estrutura primária quanto na estrutura secundária [41].

Os algoritmos de alinhamento desta ferramenta são baseados em gramáticas estocásticas livres de contexto (SCFG, *Stochastic Context-Free Grammars*) [18, 80]. Incorporada aos algoritmos de alinhamento, existe uma matriz de substituição apropriada para RNAs denominada RIBLOSUM [41], similar às matrizes usadas para proteínas, como a BLOSUM [33].

tRNAscan-SE

O programa tRNAscan-SE [53] é considerado um dos preditores de tRNAs mais precisos [46]. Ele combina três programas: dois preditores de tRNAs que buscam promotores de RNA polimerase III e características da estrutura secundária [21, 68], além de usar um Modelo de Covariância [18] treinado com sequências de tRNAs. Os dois primeiros programas são rápidos e, quando combinados, possuem uma sensibilidade superior a aproximadamente 99%. Porém, tal combinação implica em uma taxa de aproximadamente 1.85 falsos positivos por MB, o que é aceitável para genomas pequenos, mas significa aproximadamente 5.500 falsos positivos no genoma humano.

O Modelo de Covariância é bastante sensível e específico, mas muito lento. Portanto, os dois primeiros preditores de tRNAs são utilizados com baixa estringência como filtros a fim de obter candidatos promissores de tRNAs de um genoma. Os candidatos são então analisados pelo modelo de covariância, altamente estringente. O resultado é um identificador de tRNAs apresentando alta sensibilidade (99-100%) e seletividade (com uma taxa de falsos positivos inferior a 0.00007 por Mb) com uma velocidade razoável (30 Kb/s).

SVM-Portrait

O SVM-Portrait [2, 3] é adequado para identificar ncRNAs de transcritomas incompletos ou de espécies cujas caracterizações não foram concluídas. Essa ferramenta utiliza-se de métodos baseados em técnica de aprendizagem de máquina, particularmente Máquina de Vetor de Suporte (*Support Vector Machine* - SVM). O resultado do SVM-Portrait é a probabilidade de uma sequência não codificar uma proteína.

Vienna

O Vienna [36] é um pacote utilizado em pesquisas que geram ou comparam estruturas secundárias de RNAs. Esse pacote tem várias ferramentas, em que dobramentos são feitos utilizando um algoritmo de predição basendo na energia livre do RNA [105], e nas probabilidades de pareamento de bases [58].

Dentro do Vienna, o pacote RNAz, realiza a predição de estrutura baseada na energia mínima livre (MFE, *Minimun Free Energy*) [36, 106]. É levado em consideração o

fato de que as estruturas dos ncRNAs apresentam duas características: a estabilidade termodinâmica e a conservação da estrutura secundária [30]. Para o primeiro critério, o RNAz calcula uma medida normalizada da estabilidade termodinâmica e a seguir uma pontuação (z-*score*) é gerada. Uma pontuação mais negativa indica que a sequência é mais estável do que a esperada ao acaso [30].

Para o segundo critério, o RNAz prediz uma estrutura secundária consenso de um alinhamento usando a abordagem RNAalifold [35]. Mutações compensatórias (mutações que preservam um par de bases correto, por exemplo, substituição do par CG pelo par UA) são pontuadas, enquanto que mutações inconsistentes (a substituição do par CG por CA, por exemplo adicionam penalidades). No final, é calculado o índice de conservação da estrutura (SCI, *structure conservation index*) [30].

Finalmente, o RNAz utiliza um algoritmo de aprendizado de máquina SVM, que foi treinado utilizando um vasto conjunto de ncRNAs conhecidos. Esta etapa utiliza os resultados do critério (z-*score*) para classificar o alinhamento de entrada como "RNA estrutural"ou "outros" [30].

Dentro do Vienna, o pacote RNAfold [34] explora a hipótese de que uma molécula de RNA é dobrada na estrutura termodinamica mais estável, isto é, aquela que tem a energia livre mínima (ELM). Uma abordagem direta seria enumerar todas as possíveis estruturas e então selecionar aquela com o valor mínimo de energia livre [71].

RNAmmer

O RNAmmer [44] é uma ferramenta de predição de rRNAs que utiliza os bancos de dados 5S *ribosomal database* e *European ribosomal RNA database* para gerar os diversos alinhamentos estruturais necessários para à construção de bibliotecas de cadeias de Markov.

Tabela 3.2: Ferramentas computacionais

Nome	**Descrição**	**Referências**
BLAST	Compara informações de sequências biológicas primária	[1]
Infernal	Baseado em Gramática Estocástuca Livres de Contextos	[18]
tRNAscan-SE	Usa modelo de covariância na predição de tRNAs	[53]
SVM-Portrait	Identifica ncRNAs de transcritomas incompletos	[2]
Vienna	Compara estruturas secundárias de RNAs	[36]
RNAmmer	Usa cadeia de Markov na predição de rRNAs	[44]

3.0.12 Bancos de dados

Nesta seção descrevemos os bancos de dados usados para detecção de ncRNAs. Esse respositóriso são criados tanto a partir de dados experimentais quanto computacionais [90].

NONCODE

Todos os ncRNAs do NONCODE [65] foram filtrados automaticamente da literatura e do GenBank [25] e, em seguida, tratados manualmente. O NONCODE inclui quase todos os tipos de ncRNAs, exceto tRNAs e rRNAs. Mais de 80% das entradas do NONCODE

estão baseadas em dados experimentais. A primeira versão do NONCODE (v1.0) continha 5.339 sequências de 861 organismos, hoje na versão v3.0 com mais de 411.552 [49].

RNAdb

O RNAdb [77] é um banco de dados de ncRNAs de mamíferos que contém sequências e anotações de milhares de ncRNAs, mas a maioria com papéis ainda não esclarecidos [67].

miRBase

miRBase [63] é um banco de dados de microRNAs [29].

snoRNA Database

A base de dados snoRNAbase [48, 89] contém snoRNAs humanos do tipo H/ACA e C/D *box*.

snoRNAs de plantas

O *Plant snoRNA Database* contém snoRNAs de plantas [11, 88].

fRNAdb

O fRNAdb [64] integra um conjunto de outras base de dados, dentre outros o NONCODE e o RNAdb.

Rfam

O Rfam é uma base de dados curada (revisada e supervisionada), contendo informações sobre as famílias de ncRNAs. Esta base de dados consta de duas classes distintas de dados: os perfis de modelos de covariância (CMs) e os alinhamentos semente (*seed alignments*). Como dito anteriormente, os CMs são modelos estatísticos resultantes da combinação de informações tais como estrutura secundária e sequências primárias, representadas pelo alinhamento múltiplo de sequências. Cada perfil de CM corresponde a uma família de ncRNA. Já os alinhamentos semente estão contidos dentro de um arquivo no formato Estocolmo e contém os membros representativos de cada família de ncRNA gerados através de diversos alinhamentos estruturais [28].

Os arquivos Rfam podem ser obtidos do site de FTP do Sanger [75]. Nesse trabalho foi utilizado o Rfam 10.1 de Junho de 2011 com 1.973 famílias.

Tabela 3.3: Bancos de Dados

Nome	**Descrição**	**Referências**
NONCODE	Contém quase todos os tipos de ncRNAs, exeto tRNA e rRNAs	[65]
RNAdb	Contém ncRNAs de mamíferos	[77]
mirBASE	Contém microRNAs	[63]
snoRNA	Contém snoRNAs humanos do tipo H/ACA e CD/*box*	[89]
snoRNAs de Plantas	Contém snoRNAs de Plantas	[88]
fRNAdb	Contém o NONCODE e o RNAdb	[64]
Rfam	Contém famílias de ncRNAs	[75]

3.0.13 Características e Desafios na Anotação de ncRNAs

Nesta secão, primeiro discutimos aspectos importantes na anotação de ncRNAs e, em seguida, propomos uma classificação para anotação de ncRNAs

Anotação de ncRNAs

A anotação de ncRNAs envolve três principais tipos de problemas: predição de estrutura secundária [27, 35, 40], comparação de estrutura secundária [19, 36], identificação e classificação de RNAs não-codificadores [27].

Um ncRNA normalmente requer uma estrutura tridimensional específica para desempenhar sua função [84, 94]. Uma vez que a estrutura tridimensional é determinada pela estrutura secundária, a última é utilizada como aproximação no estudo da relação estrutura-função. A estrutura secundária, por sua vez, é definida pela sequência primária. Portanto, ferramentas para predizer a estrutura secundária a partir de uma sequência de RNA são úteis para estudar sua função. Quando um conjunto de RNAs homólogos é conhecido, sua estrutura secundária consenso pode ser predita com maior confiabilidade. Além disso, a conservação de domínios estruturais em diferentes espécies constitui evidência adicional de que esses domínios estão relacionados com a função específica desta sequência.

A comparação de estruturas pode servir a muitos propósitos. Por exemplo, ela pode ser utilizada para classificar um RNA como membro de uma famílias comparando sua estrutura com a estrutura consenso das várias famílias conhecidas. Além disso, se a função de um único RNA ou de uma família não é conhecida, ela pode ser inferida comparando a estrutura desse RNA (ou consenso no caso de uma família) com um banco de dados com estruturas anotadas funcionalmente. A comparação de estruturas pode também ser utilizada para detectar a ocorrência de diferentes estruturas estáveis de uma mesma molécula (o que pode indicar a presença de alterações conformacionais possivelmente relacionadas com a função do RNA), e portanto pode ser usada para predizer mutações em uma sequência de RNA que causam rearranjos na estrutura secundária ou, ainda, para comparar um conjunto de estruturas para escolher um representante.

Finalmente, predição e comparação de estruturas, além de outros tipos de análises, podem ser utilizadas para buscar ncRNAs em genomas, tanto através de buscas de RNAs homólogos a um candidato específico ou de ncRNAs em geral, incluindo novas famílias ainda desconhecidas.

Proposta de Três Principais Abordagens

Propomos uma classificação dos métodos computacionais para detecção de ncRNAs em três grupos [27, 91]:

Homologia

Predição de ncRNAs feita por meio de comparação de genomas entre duas ou mais espécies. Essas comparações dependem de bancos de dados curados, no sentido de que quanto melhores forem as anotações do banco melhores serão as predições [55].

Dois genes são ditos homólogos se descendem de um ancestral comum, e possivelmente esses genes mantenham a mesma funcionalidade herdada. Sequências homólogas podem

ser divididas em duas classes: ortólogas e parálogas. As ortólogas são sequências relacionadas por especiação, possuindo uma descendência vertical, já as parálogas são sequências relacionadas por duplicação dentro da mesma espécie ou nos ancestrais [92].

Duas ferramentas importantes para inferir homologia que usam como métrica similaridade de sequências (quanto maior a similaridade entre duas sequências, maior a chance delas terem herdado a mesma função) são: BLAST [1], uma ferramenta para detecção de ncRNAs baseada em alinhamento par a par entre as estruturas primárias das sequências; e o Infernal [1], uma ferramenta baseada em alinhamento múltiplo que considera as estruturas secundárias das sequências. O BLAST em geral não produz bons resultados, sendo O Infernal mais sensível e específico para pesquisar ncRNAs [17].

Predição de Classe

Predição de ncRNAs feita por métodos de aprendizagem de máquina.

No aprendizado supervisionado, pode-se tomar um conjunto conhecido de ncRNAs e um conjunto conhecido de proteínas, calculando características *ab initio* dessas sequências, visando criar um modelo de predição de ncRNAs. Isso confere ao modelo maior confiabilidade.

Por exemplo, o SVM-PORTRAIT [2, 3] mostra um grau de acurácia elevado para sequências de transcritomas ainda não totalmente caracterizadas.

Modelos *De Novo*

Predição de nRNAs feita por outros modelos, diferentes de *homologia* e *predição de classe*.

Por exemplo, no modelo termodinâmico, a composição e ordenação de nucleotídeos em uma molécula de RNA é responsável por sua conformação espacial. Uma investigação dessa conformação, por sua vez, resulta em um conhecimento aproximado tanto sobre a organização da molécula quanto sobre suas propriedades fisiológicas. A avaliação termodinâmica de moléculas de RNA pode ser utilizada em conjunto com regras estruturais e topológicas para inferir a estrutura secundária ativa da molécula de RNA [106].

Um RNA com uma estrutura secundária bem definida tem energia livre associada menor do que sequências com a mesma frequência de nucleotídeos, porém sem estrutura secundária definida [55]. A partir da análise da energia livre mínima de uma molécula de RNA, é possível inferir se ela tem uma conformação de estrutura secundária estável.

Capítulo 4

Sistema Multiagentes e Regras de Inferência

Neste capítulo, será feita uma breve descrição de Sistemas MultiAgentes. Na Seção 4.0.14, falaremos sobre Agentes. A Seção 4.0.15 detalha conceitos de Sistemas MultiAgentes. A Seção 4.0.16 descreve ferramentas analisadas nesse trabalho. Por fim, na Seção, 4.0.17, serão mostrados motores de inferência para raciocínio automático utilizado nesse trabalho.

4.0.14 Agentes

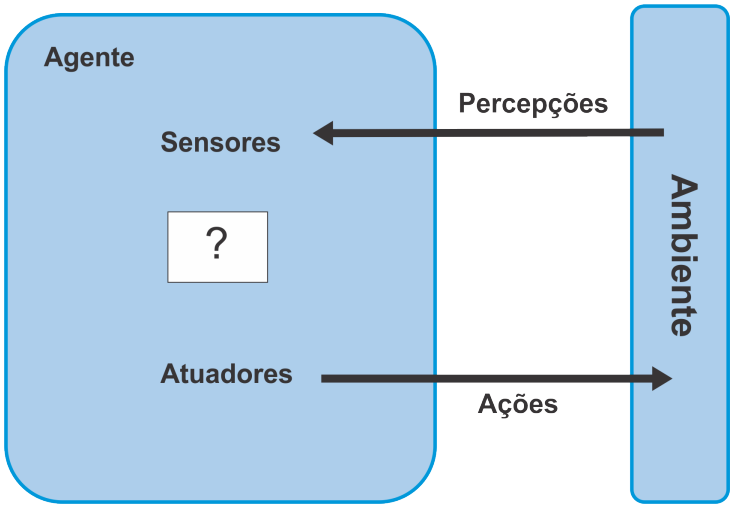

Figura 4.1: Arquitetura de um agente inteligente (Adaptado de [79]).

Um agente [79] é uma entidade capaz de perceber seu ambiente através de sensores e de agir sobre esse ambiente através de atuadores Figura 4.1. Cada agente exibe duas características fundamentais: é capaz de agir de forma autônoma tomando decisões que levam à satisfação dos seus objetivos; e é capaz de interagir com outros agentes utilizando protocolos de interação social inspirados nos humanos e incluindo pelo menos algumas das seguintes funcionalidades - coordenação, cooperação, competição e negociação [79].

4.0.15 SMAs

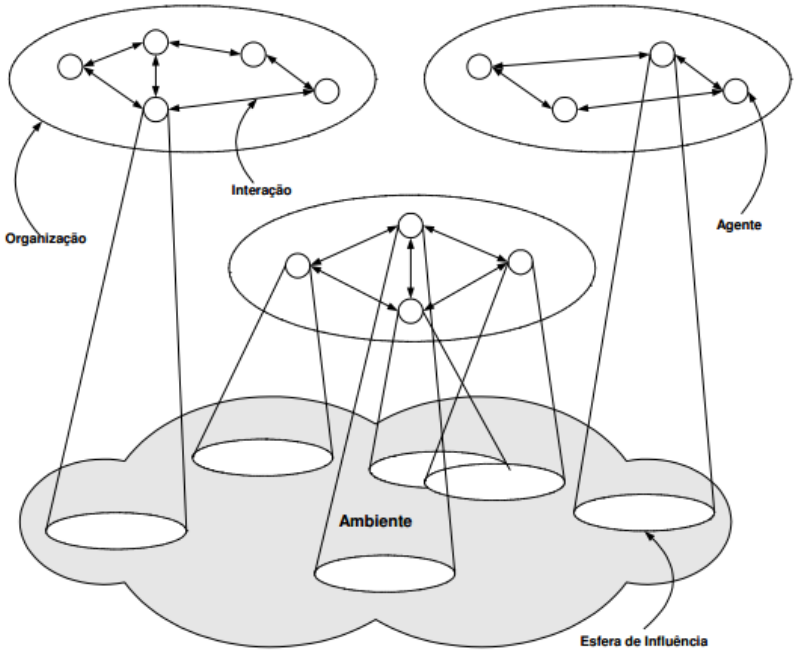

Figura 4.2: Estrutura de um Sistema Multi-Agente (Adaptado de [102]).

Os Sistemas MultiAgente (SMAs) incluem diversos agentes que interagem ou trabalham em conjunto, podendo compreender agentes homogêneos ou heterogêneos. Cada agente opera assincronamente com respeito aos outros agentes [102]. Para que um agente possa operar como parte do sistema, é necessária a existência de uma infraestrutura que permita a comunicação e/ou interação entre os agentes que compõem o SMA Figura 4.2.

4.0.16 Ferramentas para SMAs

Nesta seção, inicialmente descreveremos ferramentas para a construção de SMAs e em seguida faremos uma comparação entre elas.

Descrição das Ferramentas

Breve

É uma ferramenta livre, que permite uma criação de sistemas multiagentes 3D. Ela utiliza como linguagem de programação o Python, ou uma linguagem de *script* com o nome de Steve, ela permite definir comportamentos dos agentes em um mundo 3D e observar como eles interagem.

Cormas

É uma ferramenta livre, um framework de simulação baseado em agentes. Ela utiliza como linguagem de programação o Visual Works, que permite desenvolver para o SmallTalk, sendo orientada a objeto.

JADE

O JADE (*Java Agent DEvelopment Framework*) [7] segue os padrões da FIPA (Foundation for Intelligent Physical Agents), uma organização responsável por especificar padrões para o desenvolvimento de tecnologias baseadas em agentes inteligentes. O JADE possibilita e facilita a programação de agentes inteligentes usando Java, possui muitos atributos e características que a tornam ideal para a implementação de agentes, além de simplificar o desenvolvimento por disponibilizar um *framework* que trata a comunicação, o ciclo de vida do agente e o monitoramento da execução, entre outras.

Jadex

É um pacote construído para permitir o desenvolvimento de agentes de acordo com a FIPA e a arquitetura BDI de agentes. JADEX permite a construção de agentes de software seguindo os conceitos do modelo BDI (Belief Desire Intention);

i) Usa tecnologias como XML e Java;

ii) JADEX é projetado para facilitar a implementação de agentes em Java, e, portanto permite o reuso de várias ferramentas e bibliotecas.

Um agente JADEX é também um agente JADE e, portanto todas as ferramentas disponíveis em JADE podem ser usadas também para desenvolver agentes em JADEX. A maior parte da plataforma JADE lida com a visão externa de um agente, que não difere entre um agente JADE ou JADEX.

Zeus

É um ambiente integrado para a construção rápida de aplicações com agentes colaborativos. A documentação da ferramenta Zeus é abundante, e coloca uma forte ênfase na importância da metodologia aspecto de Zeus ("A metodologia de criação do agente é vital para o uso do Zeus"). A metodologia de Zeus usa a decomposição de quatro estágios para o mesmo desenvolvimento de agente como em nossa análise, respectivamente análise de domínio, design, realização e apoio a execução.

JAS

É uma ferramenta em Java para a simulação de agentes. Possui um mecanismo de tempo para eventos discretos, com sondas estáticas embutidas, redes neurais e algoritmos genéticos.

MADKIT

É uma ferramenta livre modular e uma plataforma escalave para multiagetens. Ferramenta que trabalha com Java e foi escrito em Java.

JASON

É uma maneira rápida de eventos discretos multiagentes que tem um núcleo com biblioteca de simulação em Java, projetado para ser a base para grandes simulações personalizadas, e também para fornecer mais funcionalidade com suficiência para simulação. MASON contém uma biblioteca de modelo e um conjunto opcional de ferramentas de visualização em 2D e 3D.

MicrosoftAgent

É um conjunto de serviços programável por software que suporta a apresentação dos personagens animados interativos. Os desenvolvedores podem usar personagens como assistentes interativos para apresentar, orientar, entreter, ou melhorar as suas páginas Web ou aplicações além do uso convencional de janelas, menus e controles.

NetLogo

É um ambiente de multi-agente livre com modelagem programável. Ele é usado por dezenas de milhares de estudantes, professores e pesquisadores do mundo todo. É de autoria de Uri Wilensky e desenvolvido no CCL.

Open Architecture Agent (OAA)

É uma ferramenta focada na construção de comunidades distribuídas de agentes, onde o agente é definido como qualquer processo de software que atenda as convenções da sociedade OAA. Um agente satisfaz este requisito, registrando os serviços que ela pode fornecer em uma forma aceitável, por ser capaz de falar a linguagem de comunicação interagente (ICL), e através da partilha de funcionalidades comuns a todos os agentes OAA.

AgentBuilder

É um conjunto de ferramentas integradas para a construção de software de agentes inteligentes. É desenvolvido pela Reticular Systems Inc., e está fundamentada no modelo de Agents BDI. Esta ferramenta é notável tanto pela alta qualidade dos seu software e do modelo conhecido.

Shell for Simulated Agent Systems (SeSAM)

SESAM (Shell para Sistemas de Agente simulado) [85] fornece um ambiente genérico para modelagem e experimentação de agente baseado em simulação. Está especialmente voltada ao fornecimento de uma ferramenta para a construção fácil de modelos complexos, que incluem interdependências dinâmica ou comportamento emergente.

SIM Agent

É uma ferramenta com uma gama de recursos para pesquisa e ensino relacionadas com o desenvolvimento de agentes que interagem em ambientes de diferentes graus e tipos de complexidade. Pode ser executado como uma ferramenta de simulação pura. Ela

foi originalmente desenvolvida para suportar uma pesquisa exploratória sobre humano, como agentes inteligentes, mas também tem sido usado para projetos de estudantes a desenvolver uma variedade de jogos e simulações interativas.

Simulation of Cognitive Agents (SimCog)

É uma plataforma genérica para SMA de simulação baseada em agentes cognitivos. Ele começou em 2001, no Departamento de Informática da Universidade de Lisboa.

StartLogo

É um ambiente de modelagem programável para explorar o funcionamento de sistemas descentralizados. Sistemas que são organizados sem um gerente, coordenado sem um gerenciador. Com StarLogo, você pode modelar, muitos fenômenos da vida real, como bandos de aves, os engarrafamentos, colônias de formigas, e economias de mercado.

Swarm

Swarm é uma plataforma para os modelos baseados em agentes - MBA, que inclui:

- Um framework conceitual para projetar, descrever e conduzir experimentos em MBA;
- implementação de software com diversas ferramentas úteis, e
- uma comunidade de usuários e desenvolvedores que compartilham idéias, softwares e experiências.

Cougaar

O Cougaar (*Cognitive Agent Architecture*) [13], uma plataforma de desenvolvimento de aplicações baseadas em agentes, foi implementada em linguagem Java e disponibiliza uma plataforma *opensource* flexível para o desenvolvimento de aplicações baseadas em agentes de diferentes tipos.

JAMA

O JAMA [96], escrito em linguagem JAVA, simplifica a construção de aplicações que exploram o conceito de agentes. Possui alto grau de desacoplamento de agentes e utiliza uma rede *peer-to-peer* que garante escalabilidade, descentralização e tolerância a falhas.

Descrição dos Critérios de Avaliação

Os critérios de avaliação aqui selecionados estão diretamente relacionados com o objetivo em que nossa opinião, no mínimo, deve-se reunir para ser considerada uma boa ferramenta de SMA. Obviamente, as ferramentas aqui avaliadas não são todos os ambientes de desenvolvimento SMA conhecidos na literatura. Algumas podem, possivelmente, obter uma melhor avaliação de acordo essa análise feita por nós. No entanto, não vão satisfazer nosso problema perfeitamente, mas podem satisfazer perfeitamente o objetivo

original para qual foram criadas. Os objetivos essenciais do desenvolvimento de ambientes são:

- Acelerar o desenvolvimento, reduzindo o esforço da programação;
- Controlar a comunicação, interação e coordenação de mecanismos;
- Permitir a implementação de sistemas relativamente complexos;
- Permitir extensibilidade de código simples;
- Fornecer apoio para a implantação e execução dos sistemas.

Escala de Avaliação

A escala utilizada visa atribuir, para cada ferramenta, um número variando entre 0 e 4, interpretados da seguinte forma:

- 4, se a ferramenta atende o critério correspondente muito bem;
- 3, se a ferramenta atende o critério correspondente bem;
- 2, se a ferramenta atende o critério correspondente moderadamente;
- 1, se a ferramenta atende o critério um pouco;
- 0, se a ferramenta não atende o critério correspondente.

Definição dos Critérios de Avaliação

1. **A metodologia atende às diferentes etapas do desenvolvimentos** - A metodologia abrange as várias fases do processo de desenvolvimento. A maioria dos autores consideram que o processo de desenvolvimento de sistemas multiagentes, consiste em quatro etapas principais: Análise, desenvolvimento, implementação e execução (implantação). Muitas vezes podemos ter essa metodologia fracamente acoplada em algumas ferramentas.
2. **Facilidade de aprendizagem** - Este critério é determinado em função de vários fatores, como a qualidade da documentação, a complexidade dos componentes e a clareza dos conceitos utilizados. O conhecimento necessário para utilizar corretamente a ferramenta, incluindo a linguagem de programação, a linguagem de comunicação entre os agentes e o protocolo de interação.
3. **Transição entre o desenvolvimento e a aplicação é simples** - Facilidade para passar do modelo para a sua implementação várias metodologias desenvolvidas são muito interessantes a nível conceitual, mas não são facilmente aplicável, olhando pelo lado dos detalhes da implementação .
4. **Flexibilidade da ferramenta para o desenvolvimento** - Flexibilidade da ferramenta é a versatilidade para o uso dos seus componentes e a sua metodologia.

5. **Comunicação entre os Agentes** - O programador não deveria ter que se preocupar com a implementação de conexões de baixo nível entre vários computadores, protocolos de comunicação, gerenciamento de segurança, sincronização, serviços de transporte de mensagens e assim por diante. Este serviço deve ser já implementado ao desenvolver.

6. **Método de Depuração** - Vários erros de coordenação e sincronização podem acontecer quando está se programando durante o desenvolvimento de SMA. A descoberta e correção desses erros podem ser muito difíceis ou mesmo impossíveis sem instrumento adequados de depuração.

7. **Apoio Gráfico ao Desenvolvimento e Implementação** - O ambiente propõe interfaces gráficas, facilitando e acelerando o desenvolvimento e implementação. Estes podem ser utilizados para criação de modelos, criação de agentes, desenvolvimento da comunicação entre os agentes, e implantação dos agentes em várias plataformas.

8. **Suporte ao Gerenciamento do SMA** - A ferramenta permite a interação com o sistema. Ele permite, por exemplo, adicionar dinamicamente, modificar ou remover agentes no sistema. O interesse deste tipo de gestão é significativo, ele pode ser muito útil para poder estudar o sistema em execução, verificando ou avaliando os níveis.

9. **Simplicidade de Implementação e Redução do Esforço necessário** - Com este critério, vários fatores devem ser levados em conta, a linguagem de progamação que a ferramenta suporta, se suporta programação orientada a objeto, muilti-threading e programação em rede, tendo vantagens significativas em detrimento de outros que não. Os componentes devem ser de fácil identificação (nome, documentação, parâmetros, etc). Além disso, as classes e os serviços disponíveis devem ser de fácil uso. A redução do esforço e necessário em termos de quantidade de código, componentes, complexidade de implementação, simplicidade de utilização dos componentes que já existem.

10. **Suporte ao Uso de Banco de Dados** - Dados de conservação e de proteção é uma tarefa técnica no nível de programação, esse processo é interessante e abstrato tanto como possíveis ferramentas para facilitar.

11. **Geração automática de Código** - Facilidade para o progresso do modelo durante sua implementação. Várias metodologias desenvolvidas são muito interessantes a nível conceitual, mas que não são facilmente aplicáveis, levando para o lado da implementação.

12. **Extensibilidade do Código** - Utilitários fornecidos pelas ferramentas, como os módulos pré-definidos, agentes ou códigos gerados, devem ser facilmente modificadas. Sendo necessárias para ser capaz de facilmente adicionar o código para as peças existentes no código.

13. **Suporte ao Desenvolvimento de Sistema Distribuído** - A possibilidade de distribuir o sistema em várias maquinas é um critério muito importante no nível da execução. A ferramenta deve também permitir uma execução simples do sistema. A execução deve ser independente do ambiente.

14. **Documentação** - É importante ter uma boa documentação e de boa qualidade, abrangendo todos os componentes da ferramenta. Além disso, é claro, concisa e não ambígua.

Outros critérios foram considerados durante essa primeira avaliação comparativa. Entre esses critérios, tivemos a metodologia utilizada para o desenvolvimento, a linguagem de comunicação entre os agentes e a linguagem de programação.

Comparação entre Ferramentas

Como base na Tabela 4.1, podemos verificar que a ferramenta JADE conseguiu o melhor resultado na avaliação.

Tabela 4.1: Avaliação das Ferramentas

	Critérios														
Ferramentas	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Total**
Breve	1	1	3	1	4	3	2	3	2	1	1	1	2	4	29
Cormas	1	2	3	1	3	4	3	2	3	1	1	1	2	4	31
Jade	3	4	2	3	4	4	3	4	2	0	0	4	4	4	41
Jadex	0	0	2	3	4	4	0	4	2	0	0	4	4	4	30
Zeus	3	1	2	1	4	2	3	3	2	2	3	2	2	4	34
JAS	0	0	2	1	2	2	0	2	2	0	0	4	4	4	22
Madkit	3	2	2	3	3	3	0	4	1	0	0	3	3	3	30
JASON	2	0	2	1	2	2	0	2	2	0	2	4	4	4	24
MicrosoftAgent	3	1	3	1	4	3	3	3	2	1	1	1	2	4	32
NetLogo	2	3	2	3	3	3	1	4	1	0	0	3	3	3	31
OAA	3	2	2	3	3	3	1	1	3	0	3	3	0	3	30
AgentBuilder	2	1	3	1	4	3	3	3	2	1	1	1	2	4	31
SeSAM	2	1	2	4	2	2	0	2	2	2	0	4	1	4	26
SIM Agent	2	2	3	1	4	3	3	3	2	1	1	1	2	4	32
SimCog	0	0	2	1	2	0	2	0	2	0	0	4	4	4	20
StartLogo	1	1	2	1	2	2	0	2	2	2	0	4	4	4	25
Swarm	2	2	3	1	4	3	3	3	2	1	1	1	2	4	32
Cougar	1	1	2	1	3	3	2	2	4	1	2	3	3	2	30
JAMA	1	2	1	1	2	1	2	1	1	1	0	1	3	1	18

4.0.17 Motores de Regras de Inferência

Como estamos insteresssados na utilização de mecanismos de raciocínio, realizamos levantamento de sistemas que permitem simulação de raciocínio, baseado em informações armazenadas em base de conhecimento.

Ferramentas

JESS

O JESS (*Java Expert System Shell*) [24], criado pela NASA em 1997, é considerado o primeiro produto de software cujo objetivo é integrar os objetos de um sistema Orientado a Objetos (OO) com regras de domínio. O JESS é um software para criação de sistemas baseados em regras e foi utilizado em sistemas construídos com as linguagens de programação C e C++.

Seu motor de inferência utiliza o algoritmo RETE [23]. O JESS é na verdade uma interface para comandos (*shell*), no qual os comandos são utilizados em uma linguagem própria que, além de permitir a criação de *scripts* de regras de interpretação, possibilita que os objetos Java sejam utilizados como ligações para os fatos do contexto.

Drools

O Drools [12] é um motor para construção de bases de conhecimento e de inferência dirigida por padrões. Foi construído para interagir com Java e o conhecimento é obtido das suas regras declarativas. Uma regra Drools tem uma ou mais condições (ou fatos) que levam a uma ou mais ações (ou consequências).

Basicamente, o motor de inferência do Drools oferece a possibilidade de utilização do método de "encadeamento para frente"e do método de "encadeamento para trás"como abordagem de investigação. O algoritmo de inferência presente no Drools é o RETE [24], como no JESS, é adaptado para sistemas orientados a objetos.

Hammurapi Rules

Hammurapi Rules [32] é um software desenvolvido em Java, cuja principal idéia, segundo seus desenvolvedores (*Hammurapi Group*), é a de apoiar o desenvolvimento de sistemas baseados em regras de uma maneira mais simples e rápida.

O motor de inferência do *Hammurapi Rules* [32] oferece a possibilidade de utilização dos dois métodos tradicionais de investigação, o "encadeamento para frente"e o "encadeamento para trás", com a implementação do tradicional algoritmo de inferência RETE [23], como no JESS e no Drools.

O Hammurapi Rules permite utilizar a própria linguagem Java para escrita das regras e, consequentemente, os próprios objetos do domínio. No Hammurapi Rules os operadores presentes na linguagem Java são os responsáveis por descrever as relações e condições presentes nas regras.

Definição dos Critérios de Avaliação

1. **Integração com a linguagem de programação** - Os três produtos apresentam algum tipo de relação com a linguagem Java. É passível de uma explicação o fato de

que o Drools e o JESS não tenham obtido a classificação máxima ("PRESENTE"). Ambos têm propósitos um pouco mais gerais do que o Hammurapi Rules. Em vista disso, o Drools e o JESS abdicam de um compromisso maior com a integração, buscando uma maior abrangência de apoio no desenvolvimento. O JESS e o Drools auxiliam a criação de sistemas baseados em regras em Java, mas só a experiência na linguagem de programação não é suficiente para o processo de desenvolvimento.

2. **Monitoração automática do contexto** - No JESS, a monitoração do contexto está presente, por meio da relação entre os objetos do domínio e o contexto. O JESS oferece uma monitoração automática com base na abordagem de "monitoração por eventos", na qual sempre que os objetos do domínio alteram, as regras são investigadas. No caso do JESS, todas as regras são investigadas sempre que um dos objetos do domínio sofre alteração. A vantagem computacional associada à monitoração por eventos não parece ser importante, pelo menos sob este aspecto. No Drools, a monitoração automática pode vir a ter melhor desempenho que o JESS, mesmo adotando a abordagem de "monitoração periódica". O número de regras e o domínio do sistema serão determinantes para uma possível medida de desempenho entre as duas abordagens.
3. **Expressão de regras de primeira ordem** - Desconsiderando o número de operadores, pode-se afirmar que a expressão do conhecimento por meio de regras de primeira ordem está presente nos três produtos.
4. **Expressão de regras de segunda ordem ou temporais** - As regras de segunda ordem ou temporais só estão presentes no Drools.

Análise dos Critérios de Avaliação

Os critérios avaliação foram estabelecidos para formalizar de maneira adequada a simulação do raciocínio dos biólogos no ncRNA-Agents. A análise desses critérios, foi definida em termos de:

- ausência do recurso: AUSENTE;
- presença parcial do recurso: PARCIALMENTE;
- presença completa do recurso: PRESENTE;

Posto isso, neste trabalho é considerado para integração com a linguagem de programação: como presente, (PRESENTE) se o produto oferecer uma maneira de implementar as regras por meio dos próprios objetos JAVA, sem *scripts* intermediários; no caso de parcial (PARCIALMENTE), o produto oferece uma maneira de implementar as regras por meio dos próprios objetos JAVA, com scripts intermediários; foi tratado como ausente (AUSENTE) o caso do produto não ofereça nenhuma maneira de implementar as regras por meio dos próprios objetos JAVA.

Além disso, para a monitoração do contexto foi considerado: Como presente (PRESENTE) se o produto oferecer as duas maneiras mencionadas de monitoração automática, sendo: monitoração periódica e monitoração por eventos; como parcial (PARCIALMENTE) se o produto oferecer pelo menos uma das duas maneiras mencionadas de

monitoração automática - monitoração periódica ou monitoração por eventos; foi tratado como ausente (AUSENTE), o caso do produto não oferecer pelo menos uma das duas maneiras mencionadas de monitoração automática - monitoração periódica ou monitoração por eventos.

Comparação de Motores de Inferência

A Tabela 4.2 traz uma comparação entre os motores de inferência, mostrando o Drools com o melhor resultado para os critérios propostos.

Tabela 4.2: Avaliação dos Motores de Inferência.

	Produtos		
Critérios	**JESS**	**Hummurapi Rules**	**Drools**
Integração com linguagem de programação	Parcialmente	Presente	Parcialmente
Monitoração automática do contexto	Parcialmente	Ausente	Parcialmente
Expressão de regras de primeira ordem	Presente	Presente	Presente
Expressão de regras temporais	Ausente	Ausente	Presente

Capítulo 5

Projeto: ncRNA-Agents

Neste capítulo, apresentamos a arquitetura de um sistema de anotação de ncRNAs baseado em SMAs. Na Seção 5.0.18, apresentaremos a arquitetura adotada no sistema ncRNA-Agents. Na Seção 5.0.19, serão apresentandos os detalhes da implementação do ncRNA-Agents.

5.0.18 Arquitetura

Observamos que o presente projeto foi inspirado no trabalho de Ralha e co-autores [73, 82], que propuseram um sistema para anotação baseado em SMAs, denominado BioAgents Figura 5.1.

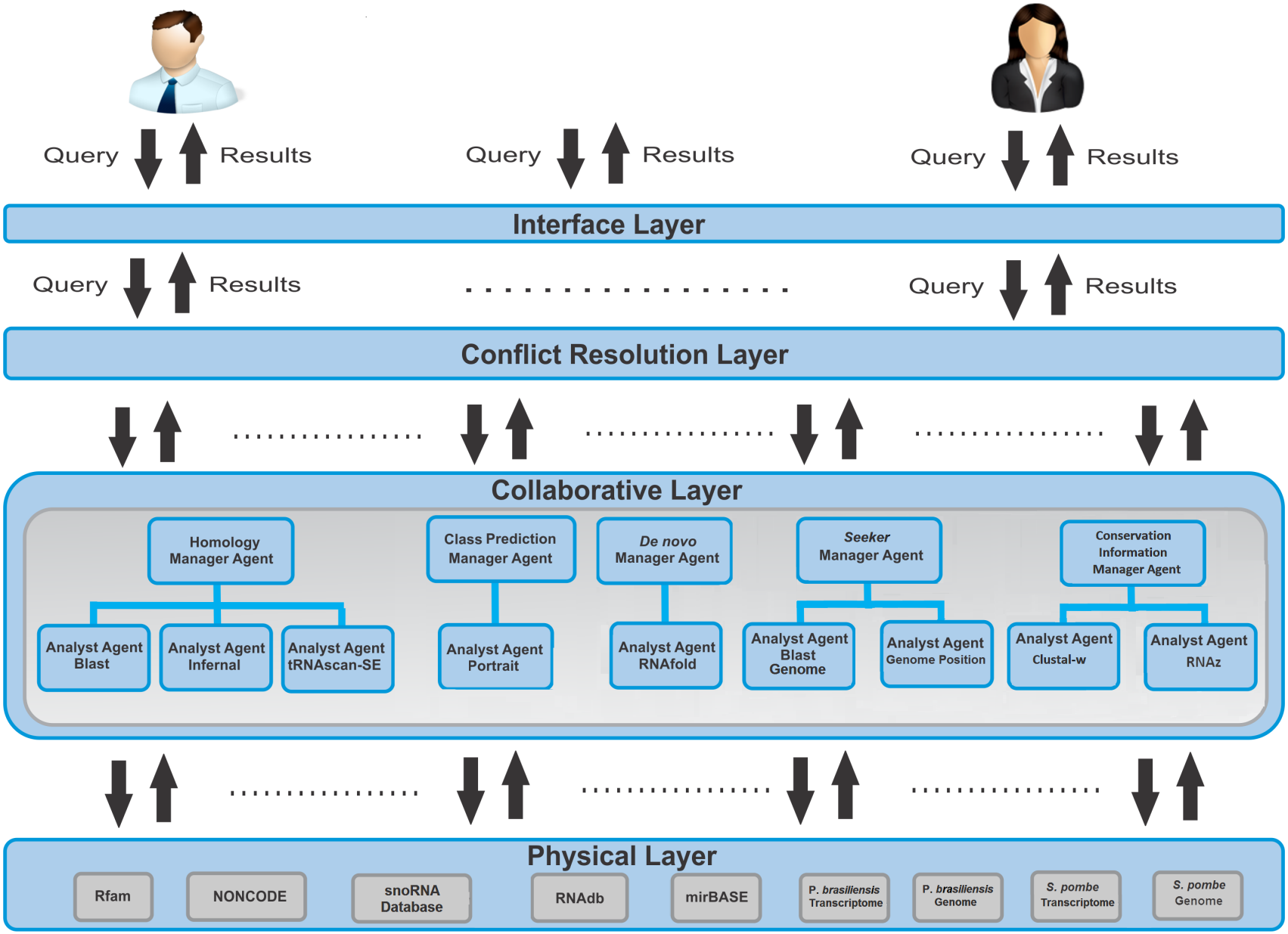

Figura 5.1: Arquitetura do ncRNAs-Agents. [4].

Descrição das Camadas

A **camada de interface** recebe a requisição do usuário composta por um arquivo com sequências em formato FASTA e indicação de ferramentas de anotação, e uma retorna os resultados de anotação das sequências de ncRNAs para o usuário. A **camada de resolução de conflitos** decide qual é a melhor recomendação para a anotação de cada sequência recebida da camada de interface, a partir das diversas sugestões recebidas da camada colaborativa, e informa essa decisão para a camada de interface. A **camada colaborativa** é responsável pela execução das diversas ferramentas para identificação e classificação de ncRNAs, enviando os resultados obtidos para a camada de resolução de conflitos. A **camada física** é formada por banco de dados.

Descrição dos Agentes

Na **camada colaborativa** existem dois tipos de agentes: **Agentes Gerentes** e **Agentes Analistas**. Os **Agentes Gerentes** realizam um filtro nas sugestões enviadas pelos Agentes Analistas. Temos três tipos de Agentes Gerentes, conforme as três abordagens descritas na seção 3.0.13. **Agente Gerente de Homologia** coordena agentes que trabalham com ferramentas baseadas em homologia. Dois exemplos são: (i) BLAST [1], que considera apenas a estrutura primária da sequência e identifica bem os snoRNAs [15]; e (ii) Infernal [17]. O **Agente Gerente Predição de Classe** coordena os agentes que trabalham com ferramentas baseadas em Aprendizagem de Máquina. Um exemplo é o SVM-Portrait [2]. O **Agente Gerente *De novo*** gerencia agentes que trabalham com ferramentas que não usam organismos de referência. Um exemplo é o RNAz [100], do pacote do Vienna, baseados em modelo termodinâmico. O **Agente Gerente Alinhamento** coordena agentes que trabalham com ferramentas de alinhamento, resultantes de métodos de comparação de sequência, com o intuito de descobrir motivos comuns ou regiões que permitam verificar se a sequência é de fato um ncRNA.

Os **Agentes Analistas** são responsáveis por executar ferramentas específicas para identificar e classificar ncRNAs. Cada Agente Analista, criado por solicitação de um Agente Gerente, executa uma análise (*parse*) para extrair informações do arquivo de saída criado pela ferramenta específica controlada por ele. O resultado dessa análise é retornado ao Agente Gerente solicitante como recomendação de anotação.

5.0.19 Detalhes de Implementação

Como prova de conceito, foi implementado um protótipo com três Agentes Gerentes e realizados dois experimentos.

Detalhes

Foi utilizada a plataforma Jade, por diversos motivos: (i) ser distribuído como software sob licença LGPL; (ii) a linguagem de programação suportada ser Java, possibilitando portabilidade; (iii) as especificações de JADE serem compatíveis com o padrão FIPA, oferecendo uma biblioteca de classes de protocolos de interação padronizados e prontas para serem instanciadas; (iv) a disponibilização da plataforma de agentes, com funcionalidades e ontologia de agentes, mecanismos transporte e parsing de mensagens; (v) oferece uma

comunicação eficiente de mensagens entre os agentes com a linguagem ACL [22], (vi) possui suporte a usuários, tendo uma comunidade grande e ativa de desenvolvedores e vasta documentação disponível para consulta.

Foi utilizado o Drools para simulação do raciocínio por meio de inferências nos agentes, pois permite programar regras de negócio declarativamente, separar e centralizar as regras de negócio de um sistema, e gerenciar regras alterando-as dinamicamente.

Resultados

Para a validação do método proposto neste trabalho, foram realizados dois experimentos baseados no Projeto *Paracoccidioides brasilienses* Genoma do (Projeto Genoma Pb) e no Projeto Genoma Guaraná. Ambos os experimentos foram executados tanto para validar o funcionamento do sistema quanto para verificar a acurácia da ferramenta.

A Figura 5.2 mostra a tela inicial e as ferramentas e bancos de dados já instalados no ncRNA-Agents.

As Figuras 6.2 e 6.3 mostram *screenshots* de *sniffers* para visualização do comportamento dos agentes do ncRNA-Agents

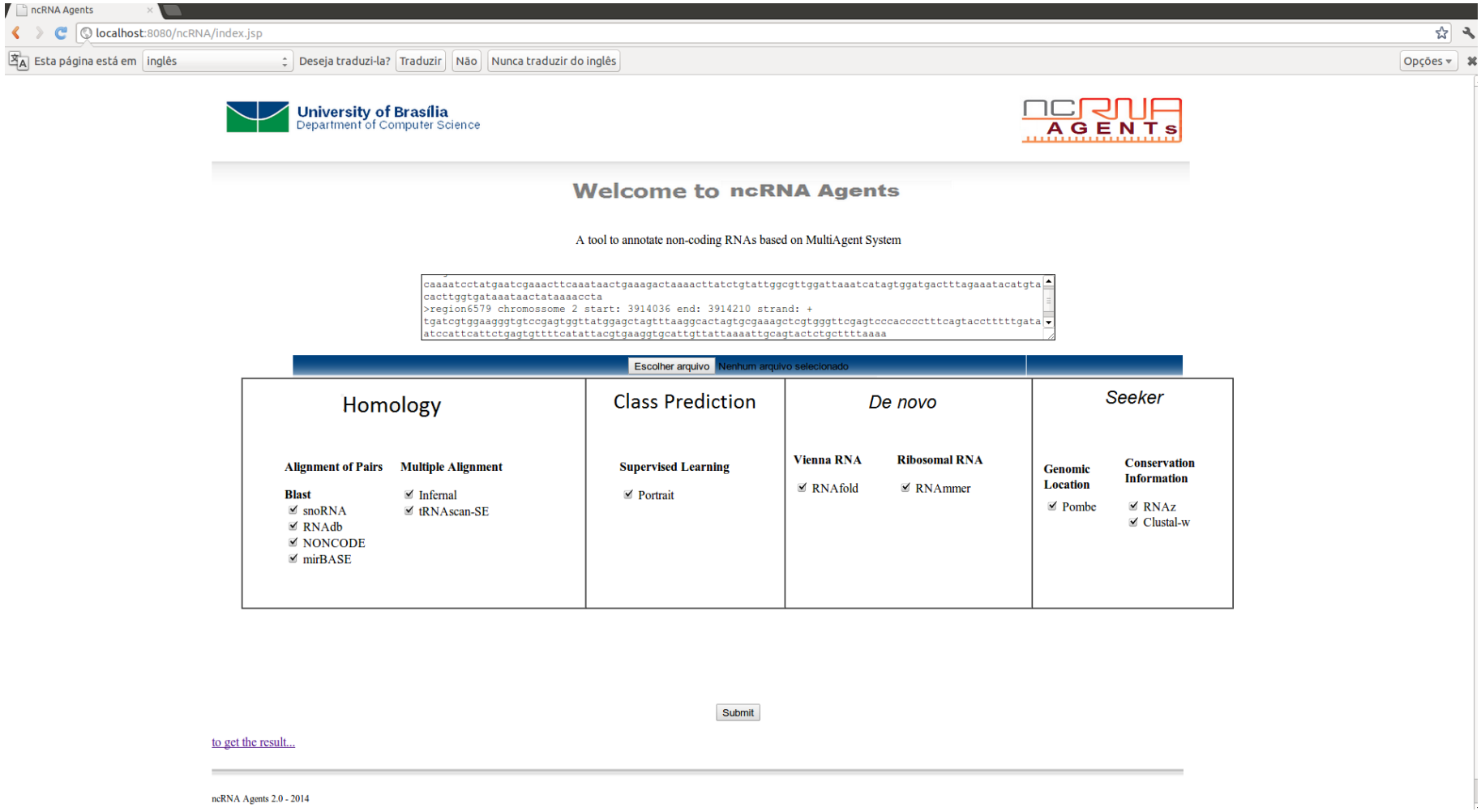

Figura 5.2: Página do ncRNA-Agents, mostrando as ferramentas e os bancos de dados usados nos experimentos.

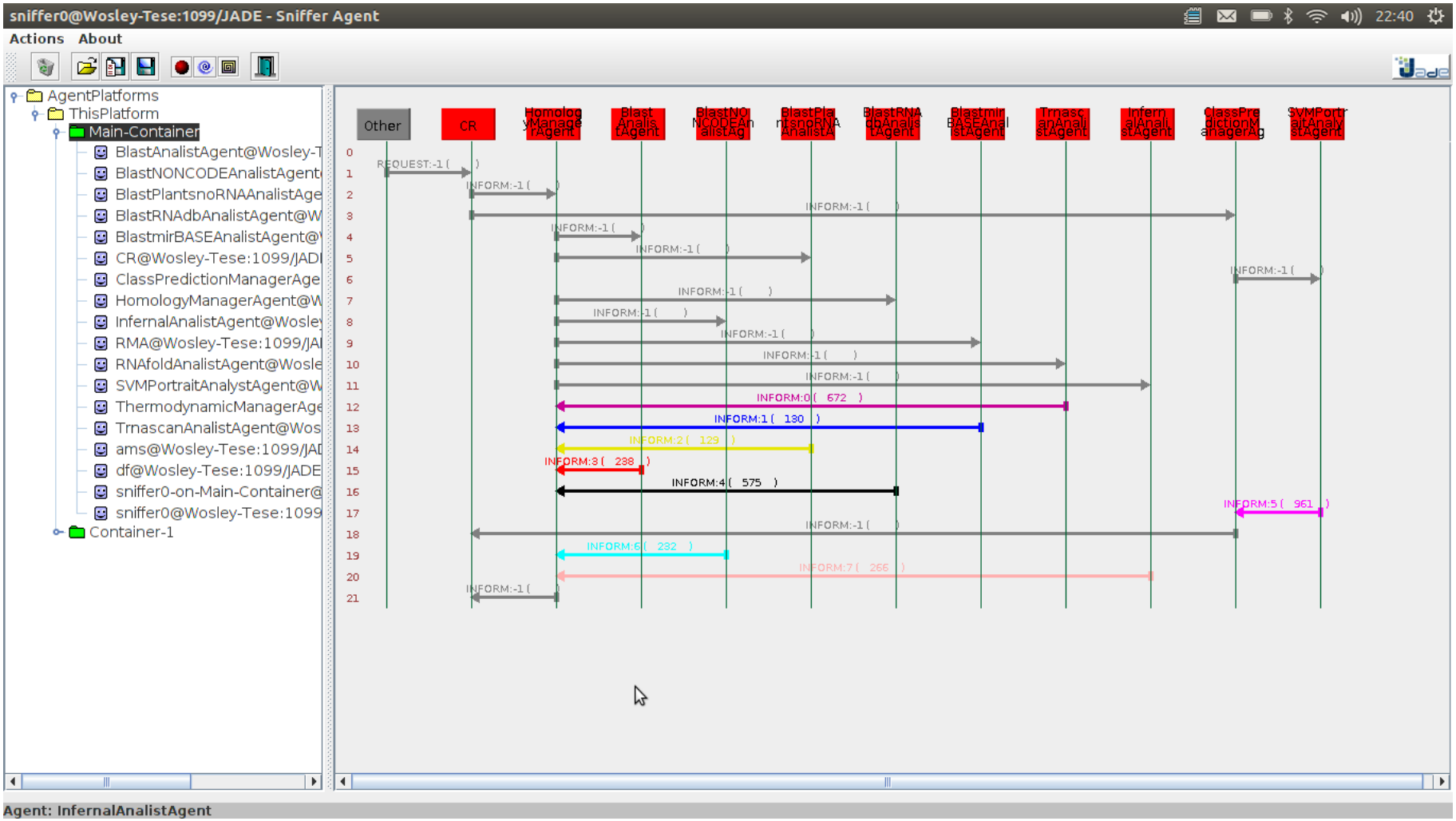

Figura 5.3: *Sniffer* dos agentes do ncRNA-Agents: a camada de resolução de conflitos aguardando a resposta do Agente Gerente Homologia com a ferramenta Infernal.

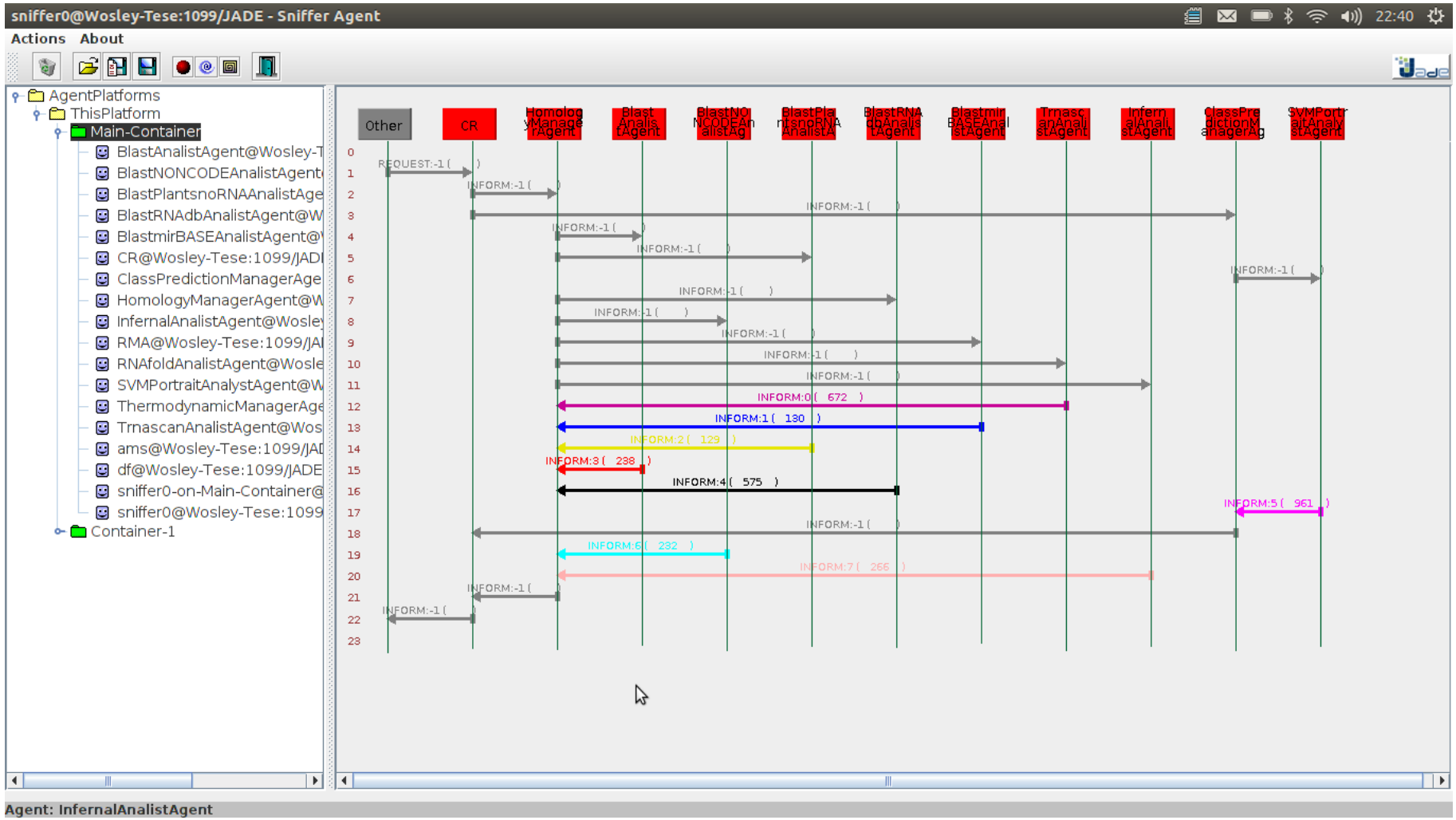

Figura 5.4: *Sniffer* dos agentes do ncRNA-Agents: a camada de resolução de conflitos enviando resposta para a interface do ncRNA-Agents para o usuário.

Obtidos

Os estudos de caso foram conduzidos para identificar ncRNAs em dois projetos transcritoma: Projeto Genoma Pb e Projeto Genoma Guaraná. No primeiro, foram utilizados: o BLAST com os bancos de dados snoRNA, RNAdb, NONCODE, mirBASE, Infernal e o banco de Rfam 10.1, trRNAscan-SE e SVM-Portrait. No Projeto Genoma Guaraná, foram usados esses mesmos bancos, além do banco de dados Plant-snoRNA.

Esses estudos foram feitos utilizando 200 sequências de cada projeto, buscando a identificação de ncRNAs, observando-se que ainda não foram identificados ncRNAs nos dois projetos. As Tabelas 6.1 e 6.2 mostram os resultados obtidos.

Tabela 5.1: ncRNAs identificados no Projeto Genoma Pb.

genome Pb		
Sequence	**Tool**	**Response**
>Contig15	SVM-Portrait	Non-Coding Probability 83.808 %
>Contig17	SVM-Portrait	Non-Coding Probability 78.872 %
>Contig24	SVM-Portrait	Non-Coding Probability 75.069 %
>Contig46	SVM-Portrait	Non-Coding Probability 91.955 %

Tabela 5.2: ncRNAs identificados no Projeto Genoma Guaraná.

genome Guaraná		
Sequence	**Tool**	**Response**
>Contig6	SVM-Portrait	Non-Coding Probability 74.163 %
>Contig29	SVM-Portrait	Non-Coding Probability 86.193 %
>Contig55	SVM-Portrait	Non-Coding Probability 81.732 %
>Contig83	SVM-Portrait	Non-Coding Probability 96.109 %
>Contig84	SVM-Portrait	Non-Coding Probability 97.258 %
>Contig93	SVM-Portrait	Non-Coding Probability 84.756 %
>Contig168	SVM-Portrait	Non-Coding Probability 78.984 %
>Contig180	SVM-Portrait	Non-Coding Probability 94.989 %

Esperados

Os testes nos dois projetos serão estendidos para todas as sequências dos dois Projetos Genoma, Pb (6.022) sequências e Guaraná (8.613).

Capítulo 6

Resultados

Para a validação do método proposto neste trabalho, foram realizados dois experimentos baseados no Projeto *Paracoccidioides brasilienses* Genoma do (Projeto Genoma Pb) e no Projeto Genoma Guaraná. Ambos os experimentos foram executados tanto para validar o funcionamento do sistema quanto para verificar a acurácia da ferramenta.

A Figura 6.1 mostra a tela inicial e as ferramentas e bancos de dados já instalados no ncRNA-Agents.

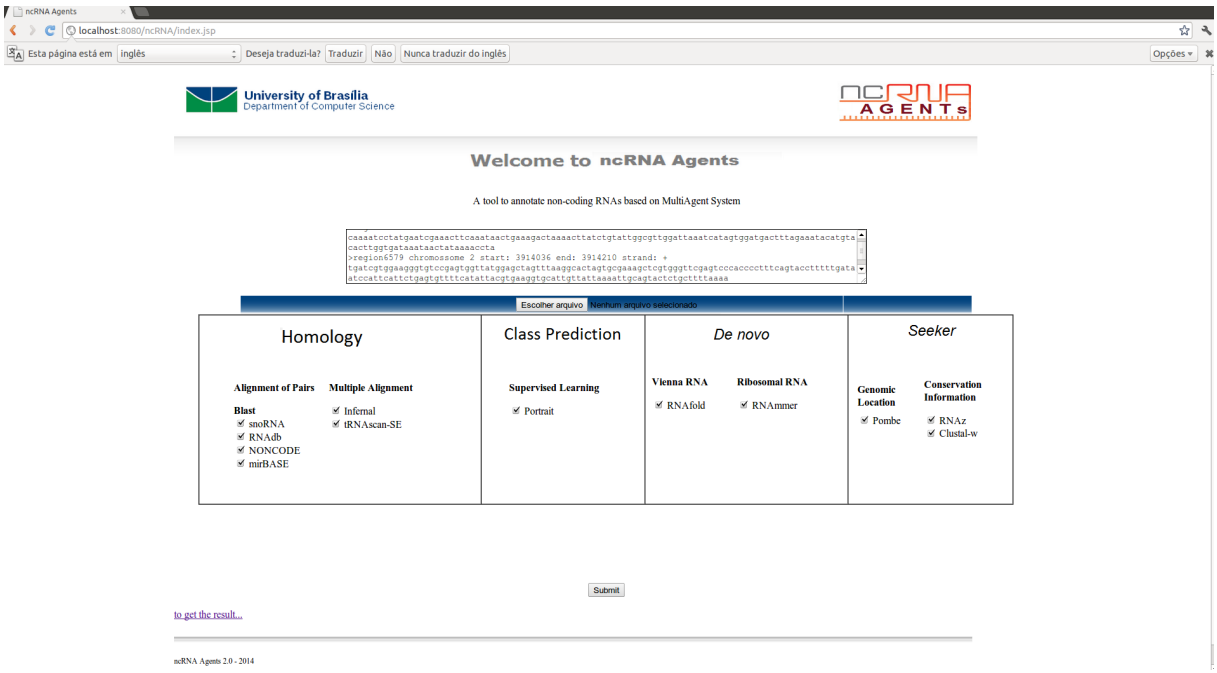

Figura 6.1: Página do ncRNA-Agents, mostrando as ferramentas e os bancos de dados usados nos experimentos.

A Figura 6.2 e 6.3 mostram *screenshots* de *sniffers* para visualização do comportamento dos agentes do ncRNA-Agents

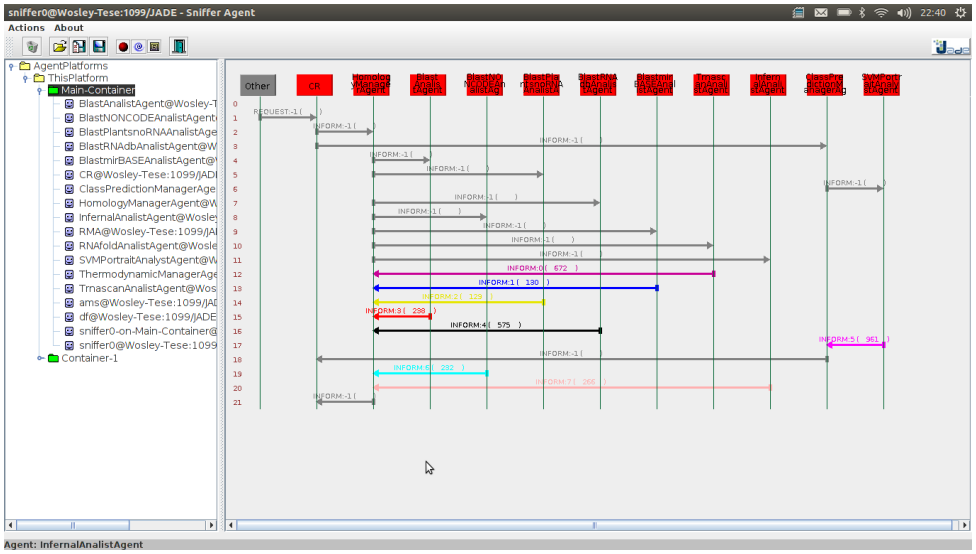

Figura 6.2: *Sniffer* dos agentes do ncRNA-Agents: Aguardando tomada de decisão da camada RC.

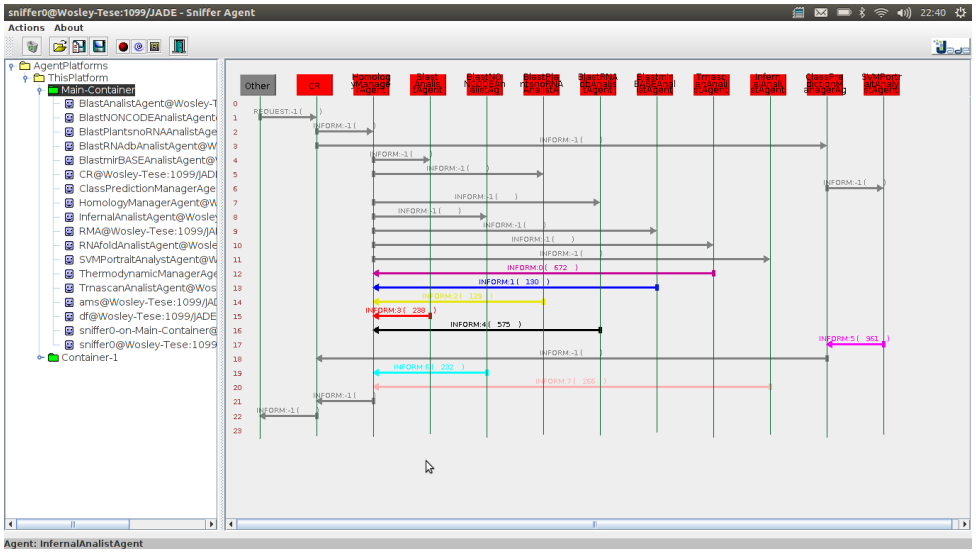

Figura 6.3: *Sniffer* dos agentes do ncRNA-Agents: Enviando resposta para a interface.

6.0.20 Obtidos

Os estudos de caso foram conduzidos para identificar ncRNAs em dois projetos transcritoma: Projeto Genoma Pb e Projeto Genoma Guaraná. No primeiro, foram utilizados: o BLAST com os bancos de dados snoRNA, RNAdb, NONCODE, mirBASE, Infernal e o banco de Rfam 10,1, trRNAscan-SE e SVM-Portrait. No Projeto Genoma Guaraná, foram usados esses mesmos bancos, além do banco de dados Plant-snoRNA.

Esses estudos foram feitos utilizando 200 sequências de cada projeto, buscando a identificação de ncRNAs, observando-se que ainda não foram identificados ncRNAs nos dois projetos. As Tabelas (6.1) e (6.2) mostram os resultados obtidos.

Tabela 6.1: ncRNAs identificados no Projeto Genoma Pb.

genome Pb		
Sequence	**Tool**	**Response**
>Contig15	SVM-Portrait	Non-Coding Probability 83.808 %
>Contig17	SVM-Portrait	Non-Coding Probability 78.872 %
>Contig24	SVM-Portrait	Non-Coding Probability 75.069 %
>Contig46	SVM-Portrait	Non-Coding Probability 91.955 %

Tabela 6.2: ncRNAs identificados no Projeto Genoma Guaraná.

genome Guaraná		
Sequence	**Tool**	**Response**
>Contig6	SVM-Portrait	Non-Coding Probability 74.163 %
>Contig29	SVM-Portrait	Non-Coding Probability 86.193 %
>Contig55	SVM-Portrait	Non-Coding Probability 81.732 %
>Contig83	SVM-Portrait	Non-Coding Probability 96.109 %
>Contig84	SVM-Portrait	Non-Coding Probability 97.258 %
>Contig93	SVM-Portrait	Non-Coding Probability 84.756 %
>Contig168	SVM-Portrait	Non-Coding Probability 78.984 %
>Contig180	SVM-Portrait	Non-Coding Probability 94.989 %

6.0.21 Esperados

Os testes nos dois projetos serão extendidos para todas as sequências dos dois Projetos Genoma, Pb (6.022) sequências e Guaraná com (8.000).

Será disponibilizado o sistema pela web.

Serão realizados mais experimentos com outros Projetos Genoma.

Capítulo 7

Conclusão

Neste capítulo descrevemos as atividades do doutorado, as etapas já realizadas e as futuras.

7.0.22 Disciplinas

7.0.23 Disciplinas

Desde o ingresso no Programa de Pós-graduação em Informática (PPGInf) da Universidade de Brasília (UnB) em 2010/2 até a data atual, foram cursadas as disciplinas, tendo sido completados 40 créditos (8 com aproveitamento de créditos).

7.0.24 Pesquisa

As atividades de pesquisa seguintes já foram concluídas:

1. Pesquisa: Levantamento do estado da arte sobre anotação de ncRNAs;
2. Estudo de SMA e Motores de Regras de Inferência;
3. Escrita de resumo sobre a ferramenta ncRNA-Agents para o WPOS-2011 [5];
4. Implementação de protótipos e estudo de caso para verificação da viabilidade do projeto.

A Tabela **??** apresenta o cronograma dos passos descritos na metodologia a ser seguida para a conclusão deste projeto:

1. Preparação e Defesa da Qualificação;
2. Implementação de protótipos;
3. Escrita de Artigos Científicos;
4. Aprimoramento da ferramenta/disponibilização na web;
5. Validação e análise dos resultados;

6. Realização de Doutorado sanduíche na Alemanha: As atividades a serem executadas nesta pesquisa seguem o cronograma apresentado na Tabela **??**. As atividades referentes ao plano de estudos no exterior serão realizadas no Grupo de Bioinformática, Departamento de Ciência da Computação da Universidade de Leipzig - Alemanha[1], sob supervisão do Professor Peter F. Stadler[2].

7. Escrita da Tese;

8. Defesa.

[1] http://www.bioinf.uni-leipzig.de/
[2] http://www.bioinf.uni-leipzig.de/ studla/

Referências

[1] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, D.J. Lipman, et al. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990. 13, 19, 21, 24, 38

[2] R. T. Arrial. Predição de RNAs não-codificadores no transcriptoma do fungo *Paracoccidioides brasiliensis* usando aprendizagem de máquina. *Dissertação de Mestrado do Departamento de Ciência da Computação da UnB*, 2006. 20, 21, 24, 38

[3] R.T. Arrial, R.C. Togawa, and M.M. Brigido. Screening non-coding rnas in transcriptomes from neglected species using portrait: case study of the pathogenic fungus paracoccidioides brasiliensis. *BMC bioinformatics*, 10(1):239, 2009. 20, 24

[4] W. C Arruda, C. G. Ralha, M. E. M. T. Walter, and P.F. Stadler. Comunicação verbal. 2011. vii, 37

[5] W. C. Arruda, C. G. Ralha, M. E. T. Walter, and I. S. Bezerra. ncrna-agents: Anotação de rnas não-codificadores baseado em sistema multiagentes. *WPOS*, 2011. 47

[6] J.H. Badger and G.J. Olsen. CRITICA: coding region identification tool invoking comparative analysis. *Molecular biology and evolution*, 16(4):512–524, 1999. 16, 17

[7] F. Bellifemine, G. Caire, A. Poggi, and G. Rimassa. JADE - a white paper. *White Paper 3, TILAB - Telecom Italia Lab. (http://jade.tilab.com/)*, v. 3:p. 6–19, 2003. 27

[8] D.A. Benson, I. Karsch-Mizrachi, D.J. Lipman, J. Ostell, and D.L. Wheeler. Genbank. *Nucleic acids research*, 34(suppl 1):D16–D20, 2006. 13

[9] M. Brain. How stuff works website - http://www.howstuffworks.co. 2000. vii, 4

[10] J. Brennecke, A.A. Aravin, A. Stark, M. Dus, M. Kellis, R. Sachidanandam, and G.J. Hannon. Discrete small rna-generating loci as master regulators of transposon activity in< i> drosophila</i>. *Cell*, 128(6):1089–1103, 2007. 18

[11] J.W.S. Brown, M. Echeverria, L.H. Qu, T.M. Lowe, J.P. Bachellerie, A. Hüttenhofer, J.P. Kastenmayer, P.J. Green, P. Shaw, and D.F. Marshall. Plant snoRNA database. *Nucleic acids research*, 31(1):432–435, 2003. 22

[12] P. Browne. *JBoss Drools Business Rules*. Packt Publishing, 2009. 33

[13] Cougaar, 2012. Cougar - http://cougaar.org. 29

[14] J.C. Dohm, C. Lottaz, T. Borodina, and H. Himmelbauer. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic acids research*, 36(16):e105–e105, 2008. 15

[15] R. Durbin, S.R. Eddy, A. Krogh, and G. Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998. 18, 38

[16] S. R. Eddy. Computacional genomics of noncoding RNA genes. *Cell*, 109:137–140, 2002. 17

[17] SR Eddy. Infernal user's guide. *http://infernal.janelia.org*, 2003. 19, 24, 38

[18] S.R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic acids research*, 22(11):2079–2088, 1994. 19, 20, 21

[19] S.R. Eddy et al. Non-coding RNA genes and the modern RNA world. *Nature Reviews Genetics*, 2(12):919–929, 2001. ix, 16, 17, 18, 23

[20] B. Ewing, L.D. Hillier, M.C. Wendl, and P. Green. Base-calling of automated sequencer traces usingphred. i. accuracy assessment. *Genome research*, 8(3):175–185, 1998. 13

[21] G.A. Fichant and C. Burks. Identifying potential tRNA genes in genomic DNA sequences. *Journal of molecular biology*, 220(3):659, 1991. 20

[22] ACL Fipa. Fipa ACL message structure specification. *Foundation for Intelligent Physical Agents, http://www.fipa.org/specs/fipa00061/SC00061G.html (30.6. 2004)*, 2002. 39

[23] C.L. Forgy. Rete: A fast algorithm for the many pattern/many object pattern match problem. *Artificial intelligence*, 19(1):17–37, 1982. 33

[24] E. Friedman-Hill. *JESS in Action*. Manning, 2003. 33

[25] Genbank, 2012. Genbank - http://www.ncbi.nlm.nih.gov/Genbank/. 21

[26] P. Green. Phrap. *Unpublished, available for download at http://www. genome. washington. edu/UWGC/analysistools/phrap. htm*, 1994. 14

[27] S. Griffiths-Jones. Annotating noncoding rna genes. *Annu. Rev. Genomics Hum. Genet.*, 8:279–298, 2007. 23

[28] S. Griffiths-Jones, A. Bateman, M. Marshall, A. Khanna, and S.R. Eddy. Rfam: an RNA family database. *Nucleic acids research*, 31(1):439–441, 2003. 22

[29] S. Griffiths-Jones, R. J. Grocock, S. Dongen, A. Bateman, and A. J. Enright. miRBase: microRNA sequences, targets and gene nomenclature. *Nucleic Acids Research*, 34:D140–D144, 2006. 22

[30] A.R. Gruber, R. Neuböck, I.L. Hofacker, and S. Washietl. The RNAz web server: prediction of thermodynamically stable and evolutionarily conserved RNA structures. *Nucleic acids research*, 35(suppl 2):W335–W338, 2007. 21

[31] R. R. Gutell, N. Larsenu, and C. R. Woese. Lessons from an evolving rRNA: 16s and 23s rRNA structures from a comparative perspective. *Microbiological Reviews*, 58(1):10–26, 1994. vii, 8

[32] Hammurapi, 2012. Hammurapi Rules - http://www.hammurapi.biz/hammurapi-biz/ef/xmenu/hammurapi-group/products/hammurapi-rules/. 33

[33] S. Henikoff and J.G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992. 20

[34] I.L. Hofacker. Vienna RNA secondary structure server. *Nucleic acids research*, 31(13):3429–3431, 2003. 21

[35] I.L. Hofacker, M. Fekete, and P.F. Stadler. Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319(5):1059–1066, 2002. 21, 23

[36] I.L. Hofacker, W. Fontana, P.F. Stadler, L.S. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte für Chemie/Chemical Monthly*, 125(2):167–188, 1994. 20, 21, 23

[37] X. Huang and A. Madan. CAP3: A DNA sequence assembly program. *Genome research*, 9(9):868–877, 1999. 14

[38] A. Hüttenhofer, M. Kiefmann, S. Meier-Ewert, J. O'Brien, H. Lehrach, J.P. Bachellerie, and J. Brosius. RNomics: an experimental approach that identifies 201 candidates for novel, small, non-messenger rnas in mouse. *The EMBO journal*, 20(11):2943–2953, 2001. 18

[39] A. Hüttenhofer, P. Schattner, and N. Polacek. Non-coding RNAs: hope or hype? *TRENDS in Genetics*, 21(5):289–297, 2005. 2

[40] B.D. James, G.J. Olsen, and N.R. Pace. Phylogenetic comparative analysis of RNA secondary structure. *Methods in Enzymology*, 180:227–239, 1989. 23

[41] R. Klein and S. Eddy. RSEARCH: finding homologs of single structured RNA sequences. *BMC bioinformatics*, 4(1):44, 2003. 20

[42] J. Koolman and K. H. Roehm. *Color Atlas of Biochemistry*. Georg Thieme Verlag, 2005. 4, 6, 7

[43] D. L. J. Lafontaine and D. Tollervey. Ribosomal RNA. *Encyclopedia of Life Sciences*, 2001. 7

[44] K. Lagesen, P. Hallin, E.A. Rødland, H.H. Stærfeldt, T. Rognes, and D.W. Ussery. Rnammer: consistent and rapid annotation of ribosomal rna genes. *Nucleic Acids Research*, 35(9):3100–3108, 2007. 21

[45] E.S. Lander, L.M. Linton, B. Birren, C. Nusbaum, M.C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001. 1

[46] D. Laslett and B. Canback. ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic Acids Research*, 32(1):11–16, 2004. 20

[47] M. Lemos, A.B. De Miranda, A.B. De Miranda, A.N.A.M. Moura, L.F.B. Seibel, M.A. CasaNova, M.V.P. De Aragao, R.N. Melo, M.A. CasaNova, M.L.Q. Mattoso, et al. Workflow for bioinformatics. 2004. 12

[48] L. Lestrade and M.J. Weber. snoRNA-LBME-db, a comprehensive database of human H/ACA and C/D box snoRNAs. *Nucleic acids research*, 34(suppl 1):D158–D162, 2006. 22

[49] C. Liu, B. Bai, G. Skogerbø, L. Cai, W. Deng, Y. Zhang, D. Bu, Y. Zhao, and R. Chen. NONCODE: an integrated knowledge database of non-coding RNAs. *Nucleic acids research*, 33(suppl 1):D112–D115, 2005. 1, 22

[50] J. Liu, J. Gough, and B. Rost. Distinguishing Protein-Coding from Non-Coding RNAs through support vector machines. *PLOS Genetics*, 2(4):529–536, 2006. 12, 17

[51] H. Lodish, A. Berk, and P. Matsudaira. *Molecular Cell Biology*. W. H. Freeman & Co, 2005. vii, 5, 6, 9, 10, 11

[52] S. Lopes. *Introdução à Biologia e origem da vida, Citologia, Reprodução e Embriologia, Histologia*. Saraiva, 1998. 5, 10

[53] T.M. Lowe and S.R. Eddy. tRNAscan-SE: a program for improved detection of transfer RNA genes in genomic sequence. *Nucleic acids research*, 25(5):0955–964, 1997. 20, 21

[54] R. L. Lundblad. *Biochemistry and Molecular Biology Compendium*. Taylor & Francis Group, 1ed edition, 2007. 9

[55] A. Machado-Lima, H.A. del Portillo, and A.M. Durham. Computational methods in noncoding rna research. *Journal of mathematical biology*, 56(1):15–49, 2008. 23, 24

[56] E.R. Mardis. Next-generation DNA sequencing methods. *Annu. Rev. Genomics Hum. Genet.*, 9:387–402, 2008. 14

[57] J.S. Mattick and M.J. Gagen. The evolution of controlled multitasked gene networks: the role of introns and other noncoding RNAs in the development of complex organisms. *Molecular Biology and Evolution*, 18(9):1611–1630, 2001. 16

[58] J.S. McCaskill. The equilibrium partition function and base pair binding probabilities for RNA secondary structure. *Biopolymers*, 29(6-7):1105–1119, 1990. 20

[59] J.T. Mendell. Micrornas: critical regulators of development, cellular physiology and malignancy. *Cell Cycle*, 4(9):1179–1184, 2005. 18

[60] T.R. Mercer, M.E. Dinger, S.M. Sunkin, M.F. Mehler, and J.S. Mattick. Specific expression of long noncoding RNAs in the mouse brain. *Proceedings of the National Academy of Sciences*, 105(2):716, 2008. 16

[61] P. Michalak. RNA world–the dark matter of evolutionary genomics. *Journal of evolutionary biology*, 19(6):1768–1774, 2006. 1

[62] F. Mignone and G. Pesole. Discrimination of Non-Protein-coding transcripts from protein-coding mRNA. *RNA biology Journal*, 2006. 17

[63] miRBase, 2012. miRBase - http://microrna.sanger.ac.uk/. 22

[64] T. Mituyama, K. Yamada, E. Hattori, H. Okida, Y. Ono, G. Terai, A. Yoshizawa, T. Komori, and K. Asai. The functional rna database 3.0: databases to support mining and annotation of functional rnas. *Nucleic acids research*, 37(suppl 1):D89–D92, 2009. 22

[65] NONCODE, 2012. NONCODE - http://www.noncode.org. 21, 22

[66] K.C. Pang, M.C. Frith, and J.S. Mattick. Rapid evolution of noncoding RNAs: lack of conservation does not mean lack of function. *Trends in Genetics*, 22(1):1–5, 2006. 17

[67] K.C. Pang, S. Stephen, M.E. Dinger, P.G. Engström, B. Lenhard, and J.S. Mattick. RNAdb 2.0? an expanded database of mammalian non-coding RNAs. *Nucleic acids research*, 35(suppl 1):D178–D182, 2007. 22

[68] A. Pavesi, F. Conterio, A. Bolchi, G. Dieci, and S. Ottonello. Identification of new eukaryotic tRNA genes in genomic DNA databases by a multistep weight matrix anaylsis of transcriptional control regions. *Nucleic acids research*, 22(7):1247–1256, 1994. 20

[69] M. Pavon-Eternod, S. Gomes, R. Geslain, Q. Dai, M.R. Rosner, and T. Pan. trna over-expression in breast cancer and functional consequences. *Nucleic acids research*, 37(21):7268–7280, 2009. 17, 18

[70] W.R. Pearson and D.J. Lipman. Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448, 1988. 13

[71] J.M. Pipas and J.E. McMAHON. Method for predicting rna secondary structure. *Proceedings of the National Academy of Sciences*, 72(6):2017–2021, 1975. 21

[72] C. Presutti, J. Rosati, S. Vincenti, and S. Nasi. Non-coding RNA and brain. *BMC neuroscience*, 7(Suppl 1):S5, 2006. 16

[73] C. G. Ralha, M. E. M. T. Walter, M. M. Brigido, and H. W. Schneider. A multi-agent tool to annotate biological sequences. *3rd International Conference on Agents and Artificial Intelligence (ICAART). Rome, Italy. The SciTePress Digital Library: The*

Institute for Systems and Technologies of Information, Control and Communication (INSTICC), pages p. 1–6., 2011. 1, 36

[74] A. Reynolds, D. Leake, Q. Boese, S. Scaringe, W.S. Marshall, and A. Khvorova. Rational siRNA design for RNA interference. *Nature biotechnology*, 22(3):326–330, 2004. 17

[75] Rfam, 2012. Rfam database - ftp://ftp.sanger.ac.uk/pub/databases/Rfam. 22

[76] E. Rivas and S. Eddy. Noncoding RNA gene detection using comparative sequence analysis. *BMC bioinformatics*, 2(1):8, 2001. 2, 17

[77] RNAdb, 2012. RNAdb - http://jsm-research.imb.uq.edu.au/rnadb. 22

[78] M. Ronaghi, M. Uhlén, P. Nyrén, et al. A sequencing method based on real-time pyrophosphate. *Science (New York, NY)*, 281(5375):363–365, 1998. 14

[79] S. J. Russell and P. Norvig. *Artificial intelligence: A Modern Approach. Second Edition.* 2002. vii, 25

[80] Y. Sakakibara, M. Brown, R. Hughey, I.S. Mian, K. Sjölander, R.C. Underwood, and D. Haussler. Stochastic context-free grammers for tRNA modeling. *Nucleic acids research*, 22(23):5112–5120, 1994. 19, 20

[81] S.L. Salzberg, A.L. Delcher, S. Kasif, and O. White. Microbial gene identification using interpolated markov models. *Nucleic acids research*, 26(2):544–548, 1998. 14

[82] H. W. Schneider. Implementação de protótipo de um SMA para anotação manual em projetos de sequenciamento de genomas. *Monografia de Graduação do Departamento de Ciência da Computação da Universidade de Brasília*, 2006. 1, 36

[83] H. W. Schneider. Método de aprendizagem por reforço no sistema BioAgents. *Dissertação de Mestrado do Departamento de Ciência da Computação da Universidade de Brasília*, 2010. 1

[84] P. Schuster, P.F. Stadler, and A. Renner. RNA structures and folding: from conventional to new issues in structure predictions. *Current opinion in structural biology*, 7(2):229–235, 1997. 23

[85] SeSam, 2012. Sesam - http://www.simsesam.de. 2, 28

[86] J.C. Setubal and J. Meidanis. *Introduction to Computational Molecular Biology.* PWS Pub., 1997. ix, 1, 16, 18

[87] N. P. Silva and L. E. C. Andrade. Noções básicas de Biologia Molecular. *Revista Brasileira de Reumatologia*, 41:83–94, 2001. 5, 9, 10, 11

[88] Plant snoRNA, 2012. Plant snoRNA database - http://bioinf.scri.sari.ac.uk/cgi-bin/plant_snorna/home. 22

[89] snoRNAbase, 2012. snoRNAbase - http://www-snorna.biotoul.fr/. 22

[90] G. Soldà, I.V. Makunin, O.U. Sezerman, A. Corradin, G. Corti, and A. Guffanti. An ariadne's thread to the identification and annotation of noncoding RNAs in eukaryotes. *Briefings in bioinformatics*, 10(5):475–489, 2009. 21

[91] P.F. Stadler. Comunicação verbal. 2011. 23

[92] L. Stryer, L. Berg, and J. L. Tymoczco. *Biochemistry*. W. H. Freeman & Co, 5ed edition, 2002. 5, 18, 24

[93] E. Torarinsson, M. Sawera, J.H. Havgaard, M. Fredholm, and J. Gorodkin. Thousands of corresponding human and mouse genomic regions unalignable in primary sequence contain common RNA structure. *Genome research*, 16(7):885–889, 2006. 17

[94] A. Torres-Larios, K.K. Swinger, A.S. Krasilnikov, T. Pan, and A. Mondragón. Crystal structure of the RNA component of bacterial ribonuclease p. *Nature*, 437(7058):584–587, 2005. 23

[95] J.C. Venter, M.D. Adams, E.W. Myers, P.W. Li, R.J. Mural, G.G. Sutton, H.O. Smith, M. Yandell, C.A. Evans, R.A. Holt, et al. The sequence of the human genome. *Science's STKE*, 291(5507):1304–1351, 2001. 1

[96] Leone. P. V. S. Vieira. Implementação de uma plataforma de desenvolvimento de sistema multiagente com abordagem peer-to-peer. *Monografia de graduação do Departamento de Ciência da Computação da Universidade de Brasília*, 2011. 29

[97] D. Voet and J. G. Voet. *Biochemistry*. Wiley, 2nd edition, 1995. ix, 12

[98] C. Wahlestedt. Natural antisense and noncoding RNA transcripts as potential drug targets. *Drug discovery today*, 11(11):503–508, 2006. 17

[99] C. Wang, C. Ding, R.F. Meraz, and S.R. Holbrook. PSoL: a positive sample only learning algorithm for finding non-coding RNA genes. *Bioinformatics*, 22(21):2590–2596, 2006. 17

[100] S. Washietl, I.L. Hofacker, and P.F. Stadler. Fast and reliable prediction of noncoding RNAs. *Proceedings of the National Academy of Sciences of the United States of America*, 102(7):2454–2459, 2005. 38

[101] J.D. Watson and F.H.C. Crick. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953. 1, 16

[102] M.J. Wooldridge. *An introduction to multiagent systems*. Wiley, 2009. vii, 2, 26

[103] Z. Xie, L.K. Johansen, A.M. Gustafson, K.D. Kasschau, A.D. Lellis, D. Zilberman, S.E. Jacobsen, and J.C. Carrington. Genetic and functional diversification of small rna pathways in plants. *PLoS biology*, 2(5):e104, 2004. 18

[104] C. Xue, F. Li, T. He, G.P. Liu, Y. Li, and X. Zhang. Classification of real and pseudo microRNA precursors using local structure-sequence features and support vector machine. *BMC Bioinformatics*, 6(1):310, 2005. 17

[105] M. Zuker and D. Sankoff. RNA secondary structures and their prediction. *Bulletin of Mathematical Biology*, 46(4):591–621, 1984. 20

[106] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic acids research*, 9(1):133–148, 1981. 20, 24