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

Métodos baseados em aprendizagem de máquina para distinguir RNAs longos não-codificadores intergênicos de transcritos codificadores de proteínas

Lucas Maciel Vieira

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

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

Brasília
2018

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

Métodos baseados em aprendizagem de máquina para distinguir RNAs longos não-codificadores intergênicos de transcritos codificadores de proteínas

Lucas Maciel Vieira

Dissertação apresentada como requisito parcial para
conclusão do Mestrado em Informática

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

Prof.ª Dr.ª Célia Ghedini Ralha CIC/UnB	Prof. Dr. André Carlos Ponce de Leon F. de Carvalho ICMC/USP - São Carlos

Prof.ª Dr.ª Bruno Luiggi Macchiavello Espinoza
Coordenador do Programa de Pós-graduação em Informática

Brasília, 1 de Março de 2018

Abstract

Non-coding RNAs (ncRNAs) constitute an important set of transcripts produced in the cells of organisms. Among them, there is a large amount of a particular class of long ncRNAs (lncRNAs) that are difficult to predict, the so-called long intergenic ncRNAs (lincRNAs), which might play essential roles in gene regulation and other cellular processes, and they can be mistaken with transcripts that code proteins. Despite the importance of these lincRNAs, there is still a lack of biological knowledge, and also a few computational methods, most of them being specific to organisms, which usually can not be successfully applied to other species, different from those that they have been originally designed to. In literature, prediction of lncRNAs performed with machine learning techniques, and lincRNA prediction has been explored with supervised learrning methods. In this context, this work proposes two methods for discriminating lincRNAs from protein coding transcripts (PCTs). The first one is a workflow to distinguish lincRNAs from PCTs in plants, considering a pipeline that includes known bioinformatics tools together with machine learning techniques, here Support Vector Machine (SVM). We discuss two case studies that were able to identify novel lincRNAs, in sugarcane (*Saccharum spp*) and in maize (*Zea mays*). From the results, we also could identify differentially expressed lincRNAs in sugarcane and maize plants submitted to pathogenic and beneficial microorganisms. The second method is the distinction of lincRNAs from PCTs using ensemble, a method that improves generalizability and robustness. We applied this method in two species, *Homo sapiens* (human), assembly GRCh38, and *Mus musculus* (mouse), assembly GRCm38. The results show good accuracies of 94% and 96% for human and mouse, respectively, which are best or at least are comparable to the accuracies presented in related works.

Keywords: long intergenic non-coding RNAs, long non-coding RNAs, non-coding RNAs, machine learning, Support Vector Machine, Ensemble

Resumo

Os RNAs não-codificadores (ncRNAs) constituem uma classe importante de moléculas produzidas nas células de organismos. Dentre eles, temos os ncRNAs longos (lncRNAs), uma classe de ncRNAs com predição díficil, pois podem estar sobrepostas a transcritos codificadores de proteínas (*Protein Coding Transcripts* - PCTs). Porém, existe uma classe de lncRNAs, os RNAs longos intergênicos (*long non-condig RNAS* - lincRNAS), que são lncRNAs que aparecem entre dois genes, que vêm sendo estudados devido a seus papéis regulatórios nos mecanismos celulares e sobretudo porque estão ligados a doenças como câncer. Apesar da importância destes lincRNAs, poucos métodos computacionais para distinção entre essa molécula e PCTs estão disponíveis. Além disso, os métodos existentes devem ser aplicados a organismos específicos, não podendo ser utilizados para distinguir lincRNAs de PCTs em espécies diferentes daquelas para as quais os modelos foram originalmente construídos. Na literatura, a predição de lncRNAs e lincRNAs vem sendo explorada com técnicas de Aprendizagem de Máquina. Neste contexto, este trabalho propõe dois métodos para discriminar lincRNAs de PCTs. O primeiro é um *workflow* para distinguir lincRNAs de PCTs em plantas, o qual utiliza ferramentas de bioinformática e Máquina de Vetores de Suporte, uma técnica de aprendizagem de máquina. O *workflow* foi aplicado em dois estudos de caso: cana-de-açúcar (*Saccharum spp*) e milho (*Zea mays*), tendo sido encontrados potenciais lincRNAs em ambos organismos. Além disso, um estudo de expressão diferencial de lincRNAs foi feito em cada estudo de caso, revelando possível interação desses lincRNAs com certos microorganismos que foram inoculados nas duas espécies de plantas. O segundo método propõe o uso de *Ensemble* para melhorar a capacidade de generalização e a robustez no método de distinguir de lincRNAs e PCTs. Este método foi aplicado em duas espécies, *Homo sapiens* (humano), montagem GRCh38, e *Mus musculus* (camundongo), montagem GRCm38. Os resultados mostram boas acurácias de 94% e 96% para humanos e camundongo, respectivamente. Deve-se notar que essas acurácias foram iguais ou melhores do que as acurácias de métodos existentes na literatura.

Palavras-chave: RNAs não-codificadores longos intergênicos, RNAs não-codificadores longos, RNAs não-codificadores, Aprendizagem de Máquina, Máquinas de Vetores de Suporte, Ensemble

Sumário

Lista de Abreviaturas e Siglas

BLAST : *Basic Local Alignment Search Tool*
Ctree : *Conditional Inference Trees*
DNA: Ácido desoxirribonucléico (*Deoxiribonucleic acid*)
KNN: (*K-Nearest Neighbor*)
lncRNA: RNA não-codificador longo (*long non-coding RNA*)
lincRNA: RNA não-codificador longo intergênico (*long intergenic non-coding RNA*)
ncRNA: RNA não-codificador (*non-coding RNA*)
PCT: Transcrito codificador de proteína (*Protein Coding Transcript*)
RF: *Random Forest*
RNA: Ácido ribonucléico (*Ribonucleic acid*)
SVM: Máquina de Vetores de Suporte (*Support Vector Machine*)

Lista de Figuras

Lista de Tabelas

Capítulo 1

Resumo da dissertação em português

Desde 1953, quando a estrutura de dupla hélice da molécula de DNA foi proposta por Watson e Crick [10], muitos projetos relacionados à investigação desta molécula foram desenvolvidos. A Biologia Molecular busca compreender as estruturas e funções de proteínas e ácidos nucleicos [11]. As proteínas são compostas por uma cadeia de moléculas (aminoácidos) que desempenham diferentes papéis em espécies vivas, como transporte de nutrientes, aceleração de reações químicas e construção de células. Os ácidos nucleicos armazenam informações moleculares essenciais para a manutenção da vida, bem como mecanismos para criação de proteínas e que também permitem a transferência de informações para outros organismos, através de processos de reprodução celular [11]. Na natureza, podemos encontrar dois tipos de ácidos nucleicos: DNA (ácido desoxirribonucleico) e RNA (ácido ribonucleico). O DNA armazena informações para gerar aminoácidos e moléculas de RNA.

Menos de 2% do material genético humano é composto por RNAs que codificam proteínas, também conhecidos como transcritos codificadores de proteína (*Protein Coding Transcripts* - PCTs). Dentre os RNAs, além daqueles que são expressos em proteínas, existem outros que não geram proteínas, mas desempenham um papel funcional importante em diversos mecanismos celulares [13]. Este último grupo é conhecido como RNAs não-codificadores (ncRNAs). Na literatura [14], os ncRNAs são classificados como: ncRNAs pequenos, que possuem características bem conhecidas e tamanhos entre 20 a 300 nucleótidos de comprimento; e ncRNAs longos (lncRNAs), que têm comprimentos acima de 200 nucleotídeos e baixa capacidade para sintetizar proteínas, sendo esses os transcritos menos conhecidos [15, 16]. Dentre as classes de lncRNAs, temos os ncRNAs longos intergênicos (lincRNAs), que são transcritos localizados em regiões intergênicas. Os lincRNAs desempenham papéis importantes na regulação de genes e em outros processos celulares [5].

Com o avanço das tecnologias de projetos utilizando sequenciamento de nova geração com o objetivo de analisar DNA, RNA e proteínas de vários organismos ao redor do mundo, um grande volume de dados biológicos foi criado [21, 22]. Em particular, os projetos transcritoma procuram analisar o conjunto completo de RNAs em um determinado organismo, enquanto aqueles que visam analisam o DNA são chamados de projetos genoma.

Parte do enorme volume de informações gerado por projetos genoma e transcritoma estão armazenados em bancos de dados contendo diversos tipos de informações biológicas. Por exemplo, o HAVANA [28] disponibiliza informações sobre lincRNAs.

Projetos que buscam descobrir as funções de lncRNAs incluem o problema de distinguir lncRNAs de PCTs, pois algumas classes de lncRNAs encontram-se sobrepostas a PCTs. Os lincRNAs constituem-se na única classe de lncRNAs que não são sobrepostos a PCTs. LincRNAs estão relacionados ao surgimento e supressão de doenças, e isso tem motivado a proposta de métodos computacionais para predição dessa classe especial de lncRNAs.

Na literatura encontramos poucos métodos computacionais para realizar esta tarefa. Em particular, vários desses métodos utilizam aprendizagem de máquina para distinguir lncRNAs de PCTs, como CNCI [27] , PLEK [26], lncRNA-MFDL [29], lncRNA-ID [30], lncRScan-SVM [31], lncRNApred [32] e Schneider et al. [33]. Em particular, o iSeeRNA [34] e o linc-SF [35] usam a aprendizagem de máquina para distinguir lincRNAs de PCTs em humanos e camundongos.

Além de poucos métodos para distinguir lincRNAs de PCTs, os disponíveis (descritos anteriormente) funcionam bem para organismos específicos (principalmente humanos e camundongos), mas, em geral, não têm boa capacidade de generalização, ou seja, não produzem bons resultados em espécies diferentes para as quais foram projetadas ou em dados diferentes dos que foram utilizados para o treinamento do modelo. Métodos para classificar lincRNAs em outras espécies, tais como plantas, podem dar suporte ao trabalho de pesquisadores e facilitar a predição das funções exercidas por lincRNAs.

As plantas são um foco de estudo importante, pois participam da manutenção da natureza, têm propriedades medicinais, além de serem utilizadas na produção de combustível e alimentos, dentre outras razões. Algumas espécies de plantas, como o milho e a cana-de-açúcar, têm uma importância particular, dado o seu amplo uso em todo o mundo e seu grande impacto econômico. Na literatura, podemos encontrar projetos que usam técnicas laboratoriais para encontrar e caracterizar lncRNAs em plantas [23, 24, 25]. Em particular, Wang et al. [23] indentificaram lincRNAs, usando uma montagem específica do milho.

Neste contexto, este trabalho propõe inicialmente um *workflow* que utiliza Máquinas de Vetores de Suporte (*Support Vector Machine* - SVM) e algumas ferramentas de bioinformática com o objetivo de distinguir lincRNAs de PCTs em plantas, os quais podem ser posteriormente validados experimentalmente. Em seguida, um outro método é proposto para distinguir lincRNAs de PCTs em humanos e camundongos, usando ensemble de métodos supervisionados, com o intuito de disponibilizar uma ferrramenta pública.

A primeira ferramenta proposta, o PlantSniffer, foi aplicada em dois estudos de caso, para o *Saccharum officinarum* (cana-de-açúcar) e para o *Zea mays* (milho). Na cana-de-açúcar, encontramos 67 lincRNAs potenciais. Além disso, investigamos lincRNAs diferencialmente expressos em bibliotecas tratadas com *Acidovorax avenae spp avenae*, o agente causal da doença da *red-streap* e duas bibliotecas de controle. No total, 46 dos 67 lincRNAs previstos foram diferencialmente expressos. Dentre eles, um foi testado em laboratório e reconhecido como um lincRNA, o qual demonstrou uma relação com o mir-408. Na cana-de-açúcar, o miR408 é um indício de que um micro-organismo é patógeno ou beneficial para a planta. Em relação ao milho, trabalhamos com transcritos obtidas do sequenciador *Illumina HiSeq*, armazenados em oito bibliotecas, quatro tratadas com *Herbaspirillum seropedicae* (duas de controle e duas inoculadas) e quatro *Azospirillum brasilense* (duas de controle e duas inoculadas), respectivamente. Nesse caso, nosso método usando SVM exibiu uma acurácia de 99%. Ainda nesse caso, investigamos a expressão diferencial dos lincRNAs preditos e obtivemos lincRNAs potenciais para serem analisados em laboratório. Um artigo foi publicado em Vieira et al. [36] e o texto completo pode ser encontrado em *http://www.mdpi.com/2311-553X/3/1/11/htm.*

A segunda ferramenta proposta, denominado LincSniffer, usa um método de aprendizagem conhecido como ensemble, que utiliza uma composição de modelos individuais para discriminar lincRNAs de PCTs. Dois estudos de caso, um para o *Homo sapiens* (humano), montagem GRCh38, e outro para o *Mus musculus* (camundongo), montagem GRCm38, foram desenvolvidos para avaliar a acurácia do método. Em geral, os modelos construídos com Ensemble apresentaram boas acurácias, melhores do que quando comparadas aos modelos individuais. No estudo de caso do *H. sapiens*, nosso modelo mostrou uma acurácia de 94% e quando comparados aos resultados obtidos de ferramentas encontradas na literatura, o LincSniffer mostrou uma precisão de 91% enquanto o iSeeRNA apresentou uma acurácia de apenas 56%. Em relação ao *M. musculus*, nosso modelo mostrou uma acurácia de 96%. Quando comparado com o iSeeRNA, que apresentou acurácia de 60,10%, o PlantSniffer mostrou acurácia de 90%. Além disso, análises de importância das características dos lincRNAs foram feitas e indicaram o comprimento de ORF e proporção de ORF relativamente ao tamanho do transcrito como importantes para a discriminação lincRNAs e PCTs. Além das ORFs, nossos testes indicaram que o número de ocorrências TCG's parece ter papel importante, o que deve ser verificado experimentalmente. O LincSniffer, testes, dados e resultados estão disponíveis no GitHub (*https://github.com/lmacielvieira/LincSniffer*).

Ambos os métodos propostos mostraram que modelos baseados em aprendizagem de máquina para discriminar lincRNAs de PCTs são úteis para indicar propriedades biológicas de lincRNAs, a serem validadas experimentalmente.

Capítulo 2

Introduction

Since the double helix structure of the DNA molecule was proposed by Watson and Crick [10], many projects related to the investigation of this molecule have been developed. Molecular biology is the field of biology that seeks to understand the structures and functions of proteins and nucleic acids [11].

Proteins are composed of a chain of molecules (amino acids) that play different roles in living species, such as transport of nutrients, acceleration of chemical reactions and construction of cells [12].

Nucleic acids store essential molecular information, as well as mechanisms for creating proteins, and also enable to transfer this information to other organisms, through cell reproduction processes [11]. In nature, we can find two types of nucleic acids: DNA (deoxyribonucleic acid) and RNA (ribonucleic acid). DNA stores information to generate various amino acids and RNA molecules. Among the RNAs, we have those that are expressed in proteins and others that do not generate proteins, but perform important functions in cellular mechanisms. This last group is known as non-coding RNAs (ncRNAs). It is well known that ncRNAs play important roles in the cell, such as chemical reactions catalyzes and various regulatory roles [13].

In the literature [14], ncRNAs are classified as: small ncRNAs, which have known characteristics and small size (20 to 300 nucleotides); and long ncRNAs (lncRNAs), which have length above 200 nucleotides and almost no capacity to synthesize proteins, these being the least known transcripts [15, 16]. Among the lncRNA classes, we have the long intergenic non-coding RNAs (lincRNAs), which are transcripts located at intergenic regions. LincRNAs play important roles in gene regulation and in other cellular processes [5].

Less then 2% of the human genetic material is composed by RNAs coding for proteins, also known as protein coding transcripts (PCTs). A large part of the RNAs have many other functions, and therefore many types of ncRNAs are known [17]. In plants, lncRNAs are not well known, althought they are involved in many important cellular processes [18]. On the other side, studies of lincRNAs in human and mouse have been developed, and most of them associate these lincRNAs with regulation in diseases, in particular, cancer [19, 20].

With the improvement of technologies for high-throughput sequencing projects with the aim of analyzing DNA, RNA and proteins of several organisms around the world, large volume of biological data were created [21, 22]. In particular, transcriptome projects seek to analyze the full set of RNAs in a given organism, while the ones that analyze DNA are called genome projects.

Plants are important focuses of study because they participate in nature maintainence, have medicine properties, are used on fuel production and as food, among other reasons. They serve as food to nearly all organisms and humans eat either plants or other organisms that eat plant. Some plant species, like maize and sugarcane, have a particular importance given their wide use around the world an their huge impact on the economy.

In plants, there are projects to find and characterize lncRNAs [23, 24, 25], relying mostly in laboratorial techniques. In particular, Wang et al. [23] also identified lincRNAs, using a specific maize assembly. Methods to predict lincRNAs in organisms (plants in specific) have to have a reference genome. Among the prediction methods present in literature, few [26, 27] discriminate lncRNAs from PCTs in plants, and they are not focused on lincRNAs.

Besides, the available methods (described previously) work well for specific organisms (mainly human and mouse), but in general, do not generalize, i.e., they do not produce good results for species different from the ones they have been designed to.

In this context, at first, this work proposes a workflow that uses machine learning and some bioinformatics tools in order to predict lincRNAs in plants aiming to indicate potential lincRNAs, which have to be further studied to find their biological rules, e.g., lincRNA association with diseases. We also propose a second method to distinguish lincRNAs from PCTs in human and mouse, using an ensemble of machine learning supervised methods.

2.1 Motivation

Researches in lncRNAs have been developed, based on their roles in important cellular processes, like gene expression and regulation [37]. Many studies suggest important functional roles for DNA transcripts that do not express proteins, presented in intergenic regions, the so-called lincRNAs [38, 39, 40, 41]. However, no methods are widely used to identify lincRNAs, although there are algorithms [34] and databases [42, 43, 44] with lincRNA information.

In one hand, despite their importance in medicine and food markets, we find few data containing lincRNA information and there are no widely used tools to distinguish lincRNAs from PCTs, which could help to understand lincRNA interactions with plant diseases as well as to isolate causes associated with them, improving plant production.

On the other hand, in human and mouse, studies related to lincRNAs and PCTs discrimination had been done, but most of them use similar workflows for prediction [45, 46]. Computational methods to distinguish lincRNAs from PCTs in human and mouse can take advantage of the amount of available data. Thus, taking advantage of these different methods working together in an ensemble method could improve accuracy and refine distinction of lincRNAs and PCTs.

2.2 Problem

There are few methods based on machine learning to discriminate lincRNAs from PCTs, being these methods specific to the species used to create the models.

2.3 Goals

The main goal is to build a model that uses machine learning to discriminate lincRNAs from PCTs.

In this work, the focus is to predict lincRNAs in plants and animals. In more details, the specific goals are:

- To propose a pipeline, using SVM models, to discriminate lincRNAs from PCTs in plants:
  - To perform case studies for sugarcane and maize;
  - To create a software, public available, for distinguishing lincRNAs from PCTs in plants.
- To devise ensemble learning models to discriminate lincRNAs from PCTs in animals:
  - To perform case studies for human and mouse;
  - To create a software, public available, for distinguishing lincRNAs from PCTs in human and mouse.

2.4 Chapters description

In Chapter 2, we first present basic concepts of molecular biology and bioinformatics. Then we describe lincRNAs, their classification and biological function.

In Chapter 3, we discuss machine learning, focusing on the methods used in this project, SVM and ensemble. Also, we present a literature review about lincRNAs prediction methods.

In Chapter 4, we present our first prediction method, called PlantSniffer. First, we present the proposed pipeline, then we show case studies in *Sorghum bicolor* (sorghum) and *Zea mays* (maize).

In Chapter 5, the LincSniffer prediction method is presented. First, we describe the method, then we show two case studies in *Mus musculus* (mouse), assembly GRCm38, and *Homo sapiens* (human), assembly GRCh38.

Finally, in Chapter 6, we conclude this dissertation and suggest future work.

Capítulo 3

LincRNAs

In this chapter, we present biological concepts about lincRNAs, which are the focus of this dissertation. In Section 3.1, we describe RNA, proteins and the central dogma of molecular biology. In Section 3.2, we briefly describe sequencers, together with bioinformatics pipelines and tools. In Section 3.3, we describe biological aspects of lincRNAs.

3.1 Molecular biology

The biological processes of regulation and structural maintainance that occur in the organisms are directed by the interaction between two group of molecules: nucleic acids and proteins. In nature, we find two types of nucleic acids, DNA (deoxyribonucleic acid) and RNA (ribonucleic acid), which play roles on protein creation and system regulation. Given the importance of these molecules in life, the field of molecular biology seeks to understand nucleic acids, as well as structures and functions of proteins [11]. This dissertation focuses on a specific group of RNAs, the lincRNAs, detailed in this chapter.

3.1.1 RNA

RNA is formed by nucleotides, consisting of phosphate, ribose and a nitrogenous base (Figure 3.1).

There are four types of nitrogenous bases composing a RNA: Adenine (A), Guanine (G), Citosine (C) and Uracil (U) [48]. The RNA nucleotides are bonded through their phosphate molecules (Figure 3.2).

Usually, the RNA is found in organisms as a single chain (single strand), different from the DNA that usually are found as a double strand, formed by chains that are complementar among themselves, with complementary pairs A/T and C/G. Even that

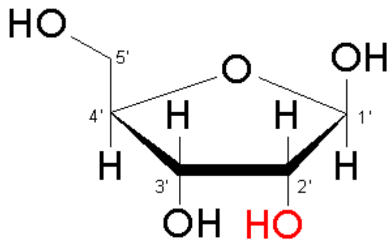


Figura 3.1: The ribose molecule is composed of five carbon atoms (1’ to 5’). Notice that carbon 2’ presents a bond with an OH molecule, which differs this molecule from deoxyribose molecule, which presents a bond with an H molecule in its carbon 2’ [47].

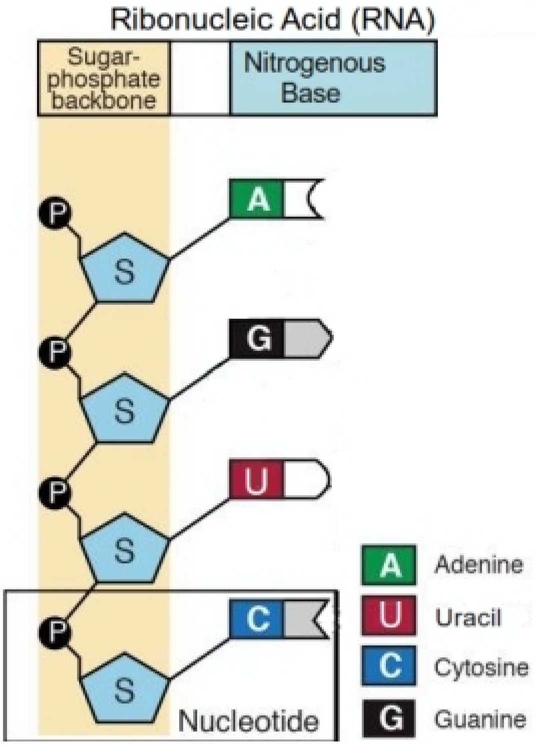


Figura 3.2: A RNA chain, bonded through phosphate molecules, composed of the four types of nucleotides present in RNA [49].

usually found as single strand, sometimes we can find hybrid DNA-RNA helices, and even RNA molecules bonded among themselves [50] (Figura 3.3).

We can find many types of RNA molecules, each one playing a different role on the cellular mechanisms [52]. Transcripts of RNAs can be divided in two groups, the protein coding (PCTs), which can be translated into proteins, and the non-coding RNAs (ncR-

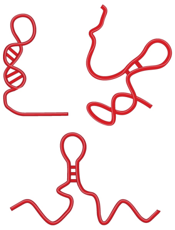


Figura 3.3: RNA strands can show bases bonded among themselves by complementarity of pairs A/T and C/G [51].

NAs), which play regulation and structural roles. As said before, in this work, we are interested in the long intergenic ncRNAs (lincRNAs), explained later.

3.1.2 The central dogma of molecular biology

The central dogma of molecular biology relates DNA, RNA and proteins, and it is divided in three processes: replication, in which a DNA strand is replicated; transcription, in which a portion of the DNA is transformed to one RNA molecule; and translation, in which two molecules of RNA are used to produce a protein (Figure 3.4).

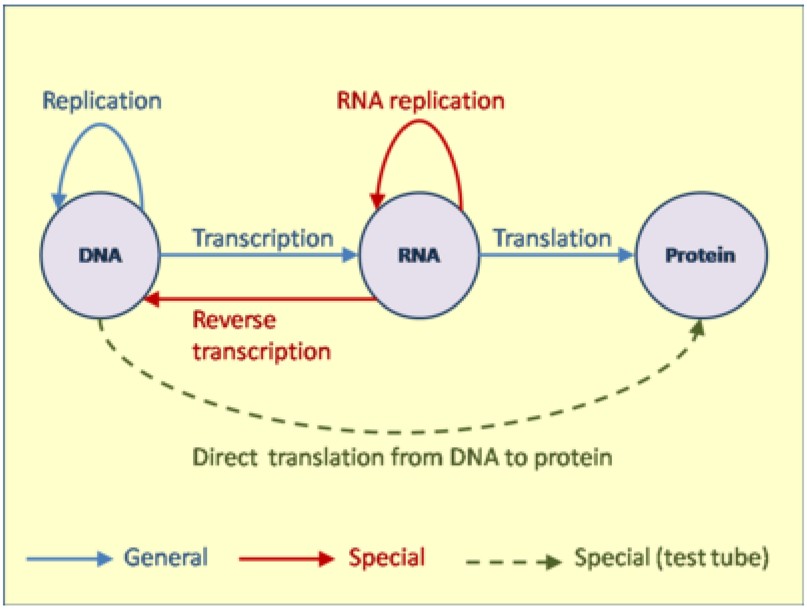


Figura 3.4: The central dogma of molecular biology, which explains the process of protein synthesis from information stored in DNA, performed with RNA molecules [53].

During the replication process, the double-stranded DNA is separated into two strands by the helicase enzyme, which binds the DNA chain and breaks the hydrogen bonds between the strands. While helicase opens the double strand, another enzyme called DNA polymerase, responsible for linking the nucleotides of the broken strands in a new complementary one, acts in parallel.

The transcription process is also initiated with the separation of the double-stranded DNA by the helicase enzyme. When the strands are separated, the RNA polymerase enzyme identifies the template strand ($5' \rightarrow 3'$) in the region of a gene (explained later). The RNA polymerase recognizes this region, which is usually preceded by a TA sequence (called TATA box) [54]. When the enzyme identifies this promoter region, the RNA polymerase guides the DNA transcription process in a not mature messenger RNA (pre-mRNA) in eukaryotes and in a messenger RNA (mRNA) in procaryotes. This DNA conversion process for RNA transcription occurs towards $5' \rightarrow 3'$, and converts the bases of the template strand to their complementary bases in the generated RNA. In Figure 3.5, we can see the difference of gene structures in eukaryotes and prokaryotes.

In eukaryotes, the pre-mRNA generated by the transcription undergoes a process known as splicing (Figure 3.6). This process removes some regions (introns) of the pre-mRNA, while binding others (exons), thus forming the mature mRNA. Note that splicing can generate more than one protein from a single gene. This process is known as alternative splicing.

After the transcription process and the splicing, the translation is started, in which the mRNA synthesizes a protein. An amino acid chain of a protein is formed in ribosomes, composed of ribosomal RNAs (rRNA), by means of a carrier, called transporter RNA (tRNA). Each tRNA binds triplets of nucleotides called codons in a tip with the corresponding amino acid on the other one (Figure 3.7).

Figure 3.8 shows the correspondence of each three bases (codon) with their corresponding amino acid, while Table 3.1 shows the 20 amino acids most commonly found in nature.

Given the genetic code, the nucleotide sequences capable of being translated into proteins, from a start codon (Methionine - AUG) to a stop codon, are called ORFs (Open Reading Frames) [11]. In Figure 3.9 we can see an example, where an ORF is translated to a protein.

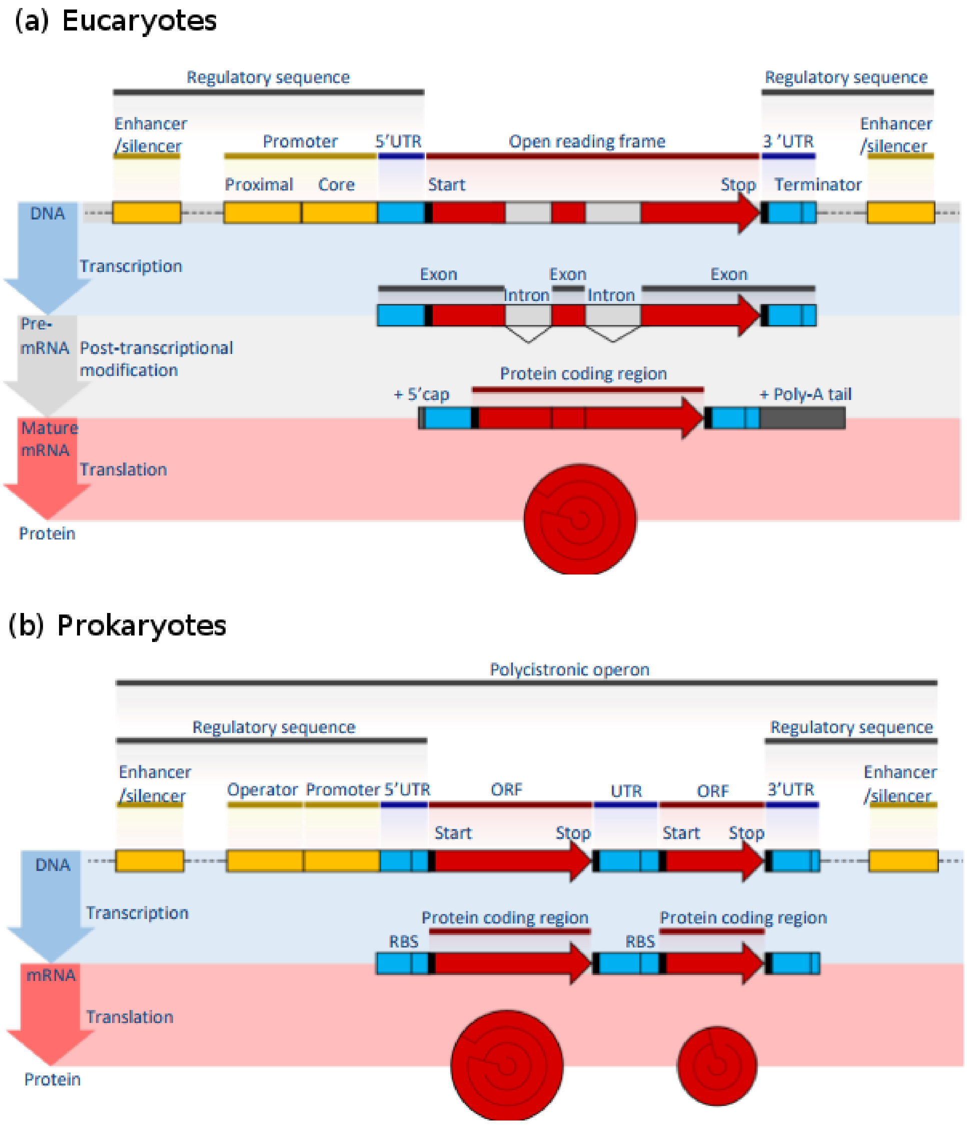


Figura 3.5: Gene structures in (a) eukaryotes and (b) prokaryotes. In eukaryotes the transcription processes generate a pre-mRNA that passes through a post-transcriptional modification in order to generate the mature mRNAs, while on prokaryotes the transcription processes do not generate the pre-mRNAs, but the mRNA itself [1].

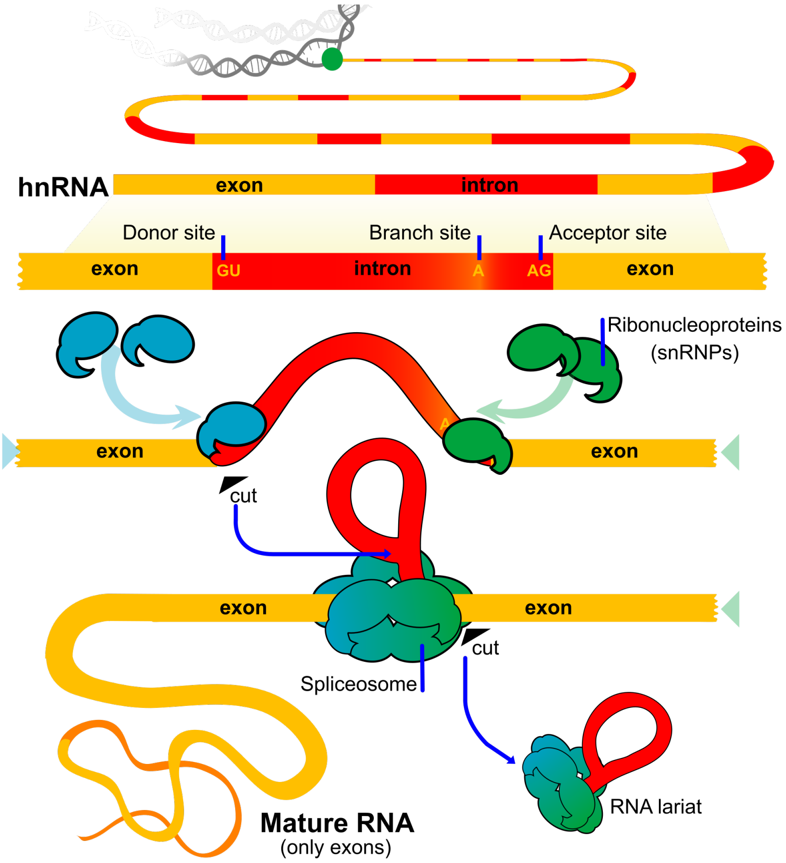


Figura 3.6: Process of *splicing* in eukaryotes. The *splicing* is the post-transcriptional process that transforms the premRNA transcript into a mRNA by removing the introns and joining the exons. We can note that some different types of ncRNAs are involved in the process. This processes can create a variety of different mRNAs from pre-mRNAs, being this phenomenon called alternative splicing [2].

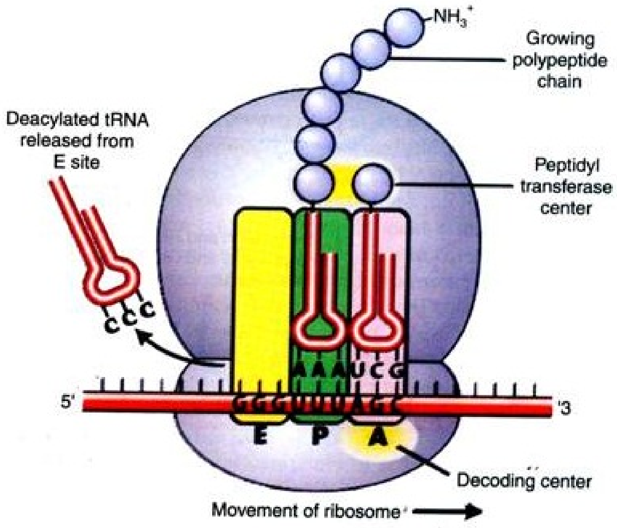
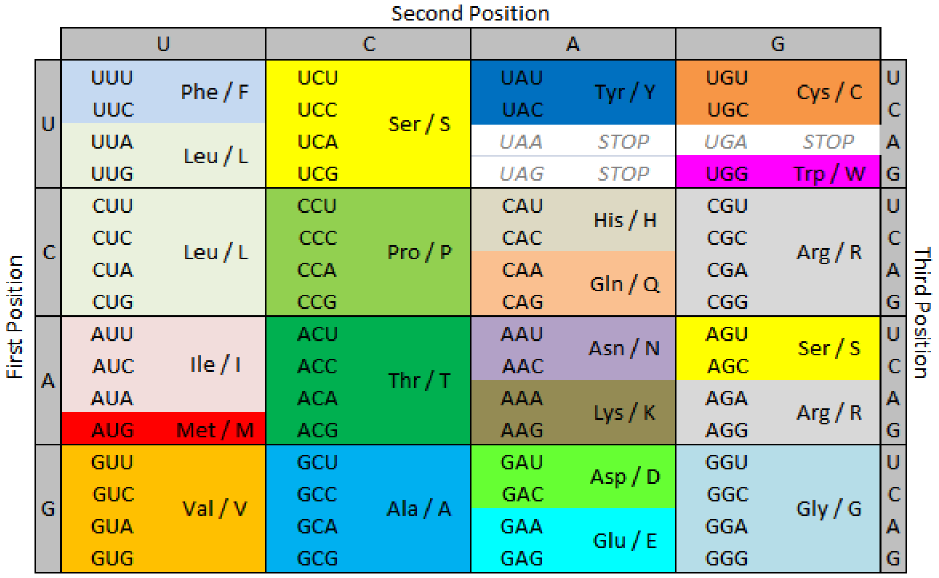

NH_3^+
Growing polypeptide chain
Deacylated tRNA released from E site
Peptidyl transferase center
CCC
AAAUCG
5'
GGGUUUAGC
'3
E
P
A
Decoding center
Movement of ribosome

Image 4.8. Translation-making protein

Figura 3.7: Translation, noting that two molecules of ncRNAs are involved in the process, rRNA and tRNA. The translation processes transforms mRNAs into proteins by translating each RNA tiplet (codon) to its correspond amino acid, which will form a chain (called polypeptide) and therefore a protein [3].



		Second Position								
First Position		U		C		A		G		Third Position
	U	UUU	Phe / F	UCU	Ser / S	UAU	Tyr / Y	UGU	Cys / C	U
		UUC		UCC		UAC		UGC		C
		UUA	Leu / L	UCA		UAA	STOP	UGA	STOP	A
		UUG		UCG		UAG	STOP	UGG	Trp / W	G
	C	CUU	Leu / L	CCU	Pro / P	CAU	His / H	CGU	Arg / R	U
		CUC		CCC		CAC		CGC		C
		CUA		CCA		CAA	Gln / Q	CGA		A
		CUG		CCG		CAG		CGG		G
	A	AUU	Ile / I	ACU	Thr / T	AAU	Asn / N	AGU	Ser / S	U
		AUC		ACC		AAC		AGC		C
		AUA		ACA		AAA	Lys / K	AGA	Arg / R	A
		AUG	Met / M	ACG		AAG		AGG		G
	G	GUU	Val / V	GCU	Ala / A	GAU	Asp / D	GGU	Gly / G	U
		GUC		GCC		GAC		GGC		C
		GUA		GCA		GAA	Glu / E	GGA		A
		GUG		GCG		GAG		GGG		G

Figura 3.8: Triplets of RNA (codons) are translated in amino acids. This table is known as the genetic code [55].

Tabela 3.1: The twenty amino acids most commonly found in nature [11].

Abbreviation	Name
Ala	Alanine
Cys	Cysteine
Asp	Aspartate
Glu	Glutamate
Phe	Phenylalanine
Gly	Glycine
His	Histidine
Ile	Isoleucine
Lys	Lysine
Read	Leucine
Met	Methionine
Asn	Asparagine
Pro	Proline
Gin	Glutamine
Arg	Arginine
Ser	Serina
Thr	Threonine
Val	Valine
Trp	Tryptophan
Tyr	Tyrosine

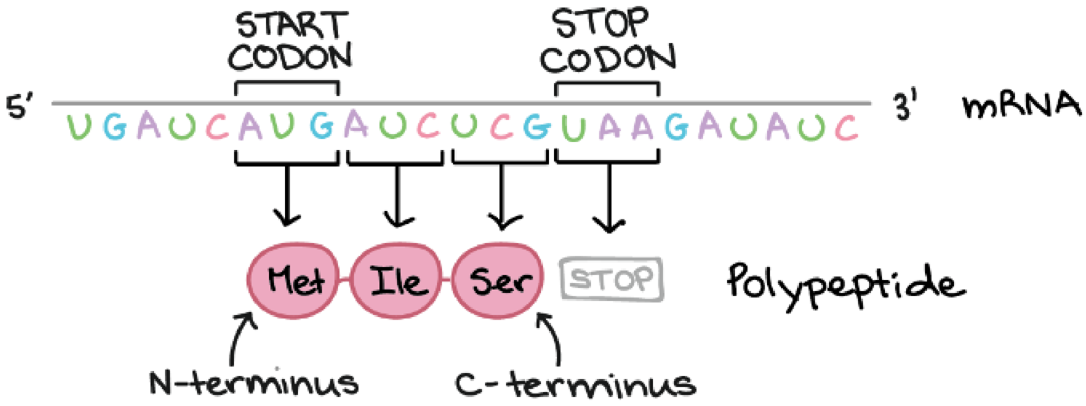


Figura 3.9: The mRNA represented by UGAUCAUGAUCUCGUAAGAUAUC, where the strand goes from 5' to 3', and at the sixth base we can find the start of the triplet AUG, the start codon. From the start codon until the fifteenth base pair, which represents the stop codon UAA, we have two triplets (AUC and UCG), that are translated into Isoleucine and Serina and result into a protein [56].

3.2 Sequencing and bioinformatics pipeline

In this section, we briefly describe sequencing technologies and after, we show how bioinformatics pipelines are constructed.

3.2.1 High-throughput sequencing

Sequencing is the process of obtaining a sequence of nucleotides that composes a given portion of DNA or RNA. The new technologies, known as high-throughput sequencing, have evolved very fast in the last years. These technologies perform the DNA sequencing in platforms capable of generating millions of bases in a short period of time. Currently, the Illumina sequencer [57], which performs sequencing by synthesis, is one of the most used.

The sequencing process of Illumina starts when the DNA to be sequenced is received. At first, the received DNA is fragmented and bonded to adapters at their 5' and 3' ends. Next, the DNA molecules are bound to a solid support, where there are oligonucleotides complementary to the adapters on the ends of the molecules.

When connected to the supports, the DNA amplification step occurs, by using the Polymerase Chain Reaction (PCR) technique. The PCR uses an enzyme known as *Taq DNA polymerase* to replicate the DNA strands, in which the molecules that are attached to the support are amplified. This amplification process is repeated until that many groups of identical molecules are formed on the support plate.

With enough DNA molecules and a labeled terminator incorporated[1], a laser excitement is done, in order to generate a light signal, which differs from terminator to terminator. This signal is picked up by a reading device and interpreted as one of the four core components of nucleotides molecules.

The process terminator merging, excitement and reading is repeated for each nucleotide that composes the sequence until the final sequencing is produced [4]. Figure 3.10 shows the sequencing process of Illumina.

[1] A sequence of pre-determined nucleotides.

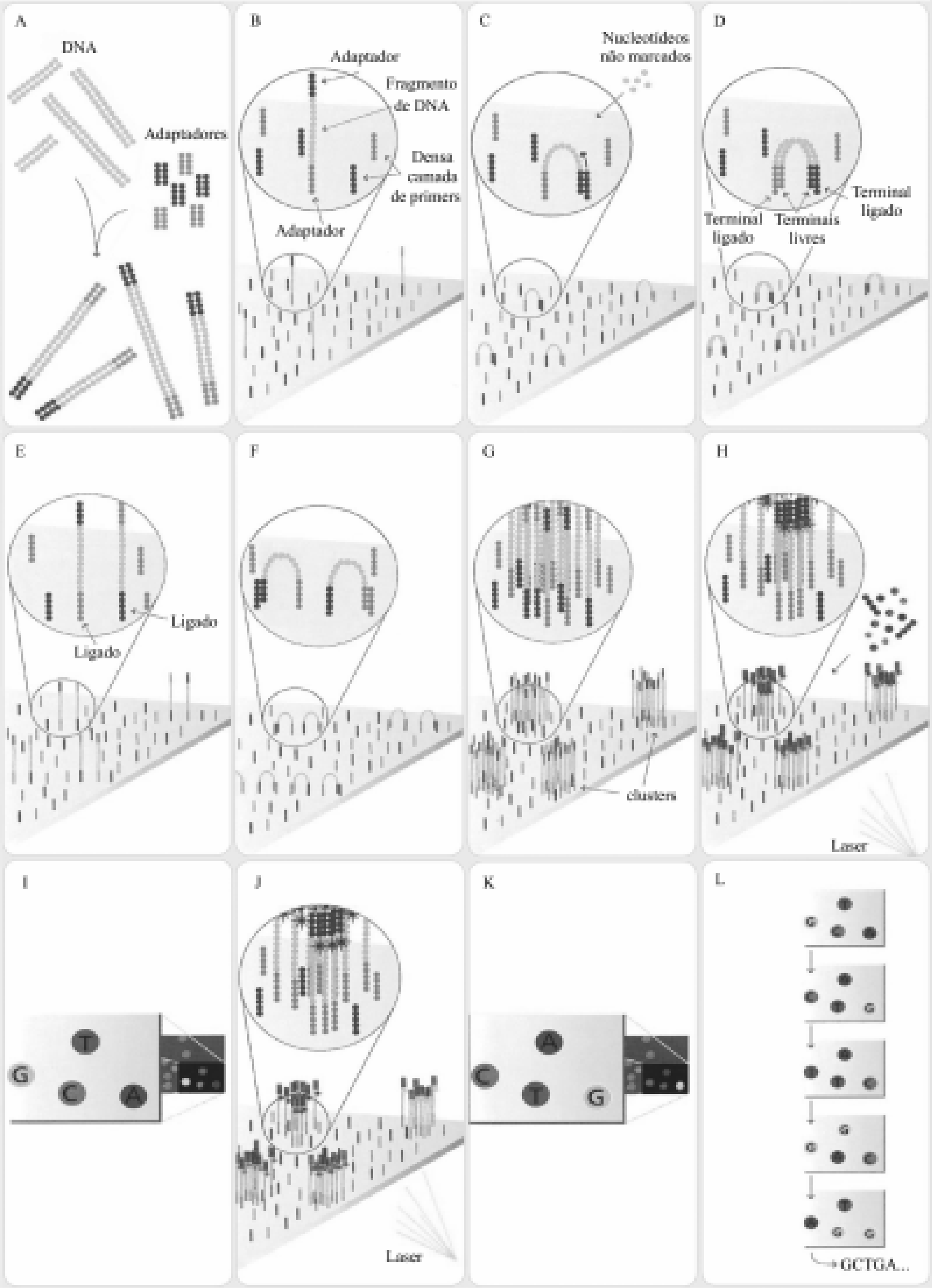


Figura 3.10: Sequencing process used by *Illumina* [4].

Besides Illumina, there are other sequencing technologies and profiling methods, i.e, *DNA nanoball sequencing* [58] and *Helioscope single molecule sequencing* [59]. A common used transcriptome profiling method is the RNA-seq, which is used in order to analyze RNA and can be applied together with other sequencers, e.g, Illumina. RNA-seq uses deep-sequencing technologies, also providing a more precise measurement of levels of transcripts and their isoforms[2] than other methods [60]. Figure 3.11 shows a RNA-seq experiment.

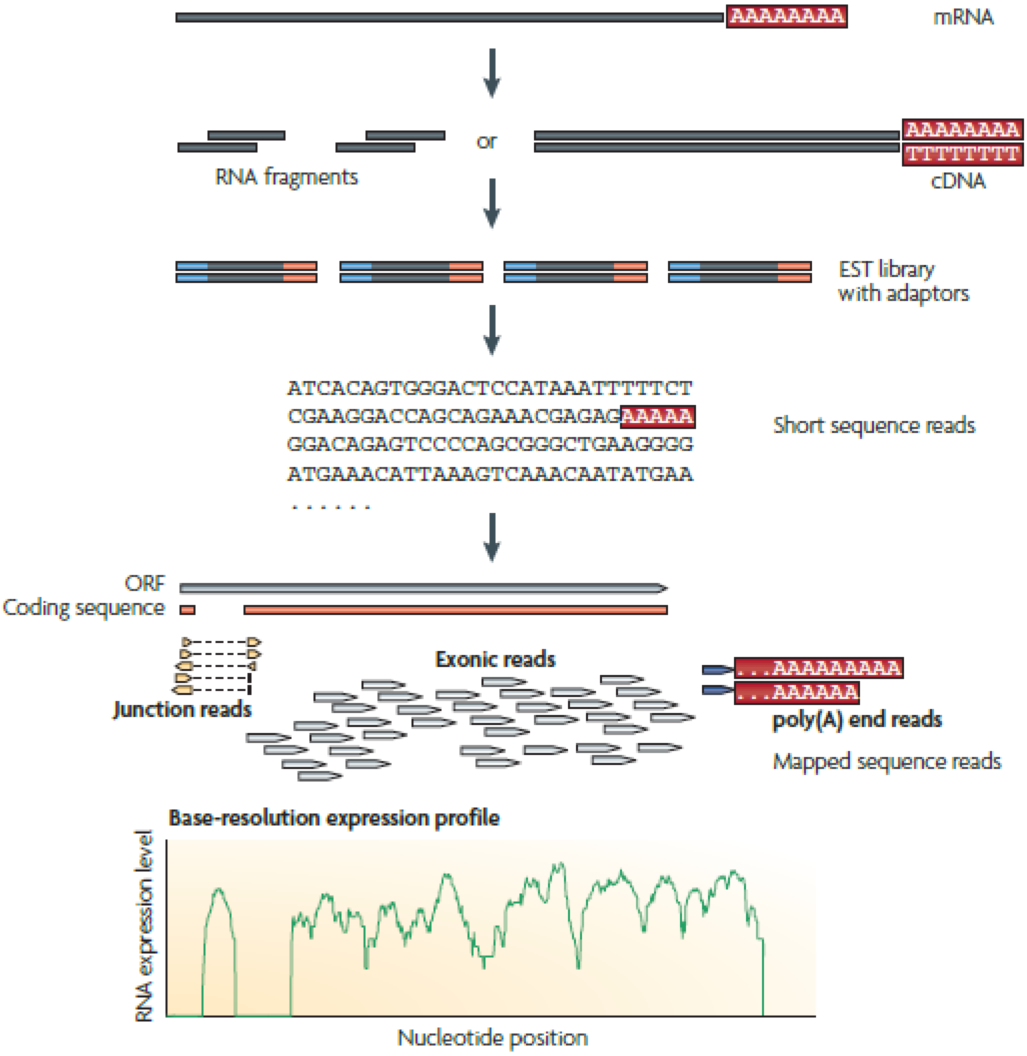


Figura 3.11: A typical RNA-seq experiment where long RNAs are converted into a library of cDNA fragments through either RNA fragmentation or DNA fragmentation. Then sequencing adaptors (blue) are subsequently added to each cDNA fragment, and a short sequence is obtained from each cDNA using high-throughput sequencing technology. The resulting sequence reads are aligned with the reference genome or transcriptome, and classified as three types: exonic reads, junction reads and poly(A) end-reads. These three types are used to generate a base-resolution expression profile for each gene, as illustrated at the bottom [60] .

[2] Different proteins (or variation of one protein) coded by same gene, through the alternative splicing process.

3.2.2 Bioinformatics pipelines

Bioinformatics is an area where researchers aim to create and apply computational and mathematical techniques to analyze information generated by sequencing projects [61]. In order to analyze these DNA and RNA sequences, we use workflows, particularly, pipelines.

A pipeline is defined by a sequence of computational methods used to treat the data generated by a transcriptome or a genomic project, where the output files of one step of the pipeline is used as input for the next step. An example of pipeline can be seen in Figure 3.12.

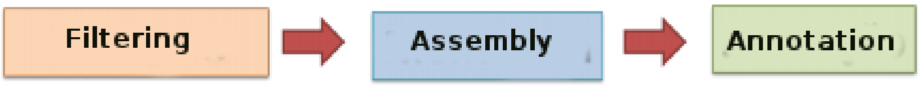
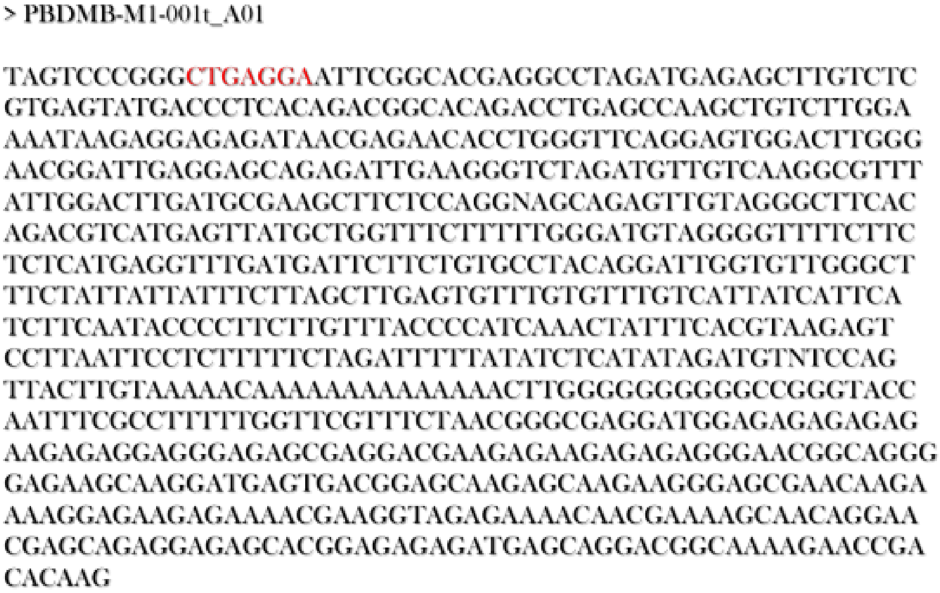


Figura 3.12: Example of a pipeline, with three steps.

As said before, in sequencers, the DNA/RNA sequences are transformed in character chains over the alphabet $\sum = \{A, C, G, T/U\}$. These sequences are stored in files with well known and defined formats, as *fasta* and *fastq*. *Fasta* is one of the most used formats, and it is defined by having its first line started with the character '>', which indicates the identifier of a sequence, followed by other lines that show the charactes in the genome/transcript sequence (Figure 3.13).

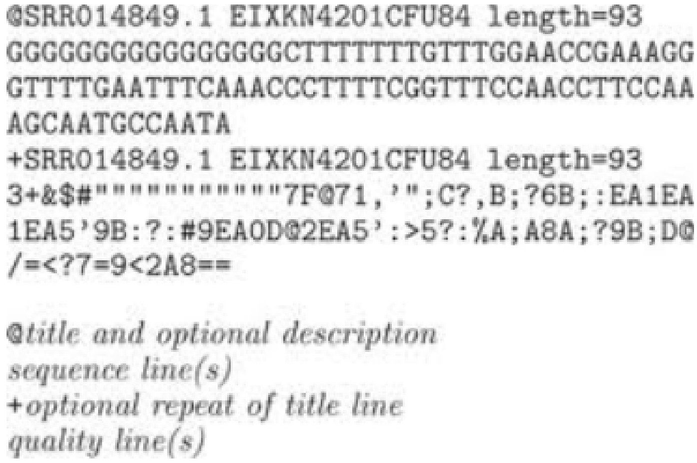


```
> PBDMB-M1-001t_A01

TAGTCCCGGGCTGAGGAATTCGGCACGAGGCCTAGATGAGAGCTTGTCTC
GTGAGTATGACCCTCACAGACGGCACAGACCTGAGCCAAGCTGTCTTGGA
AAATAAGAGGAGAGATAACGAGAACACCTGGGTTCAGGAGTGGACTTGGG
AACGGATTGAGGAGCAGAGATTGAAGGGTCTAGATGTTGTCAAGGCGTTT
ATTGGACTTGATGCGAAGCTTCTCCAGGNAGCAGAGTTGTAGGGCTTCAC
AGACGTCATGAGTTATGCTGGTTTCTTTTTGGGATGTAGGGGTTTTCTTC
TCTCATGAGGTTTGATGATTCTTCTGTGCCTACAGGATTGGTGTTGGGCT
TTCTATTATTATTTCTTAGCTTGAGTGTTTGTGTTTGTCATTATCATTCA
TCTTCAATACCCCTTCTTGTTTACCCCATCAAACTATTTCACGTAAGAGT
CCTTAATTCCTCTTTTTCTAGATTTTTATATCTCATATAGATGTNTCCAG
TTACTTGTAAAAACAAAAAAAAAAAAAAACTTGGGGGGGGGGCCGGGTACC
AATTTCGCCTTTTTGGTTCGTTTCTAACGGGCGAGGATGGAGAGAGAGAG
AAGAGAGGAGGGAGAGCGAGGACGAAGAGAAGAGAGAGGGAACGGCAGGG
GAGAAGCAAGGATGAGTGACGGAGCAAGAGCAAGAAGGGAGCGAACAAGA
AAAGGAGAAGAGAAAACGAAGGTAGAGAAAACAACGAAAAGCAACAGGAA
CGAGCAGAGGAGAGCACGGAGAGAGATGAGCAGGACGGCAAAAGAACCGA
CACAAG
```

Figura 3.13: Example of a *fasta* file.

Other well known format is the *fastq*, which besides having information of the characters of the genome/transcript sequence, its identifier and its description, also contains information of the qualities of each nucleotide, represented with the ASCII code, as shown in Figure 3.14.



```
@SRR014849.1 EIXKN4201CFU84 length=93
GGGGGGGGGGGGGGGCTTTTTTTGTTTGGAACCGAAAGG
GTTTTGAATTTCAAACCCTTTTCGGTTTCCAACCTTCCAA
AGCAATGCCAATA
+SRR014849.1 EIXKN4201CFU84 length=93
3+&$#"""""""""""7F@71,’";C?,B;?6B;:EA1EA
1EA5’9B:?:#9EA0D@2EA5’:>5?:%A;A8A;?9B;D@
/=<?7=9<2A8==
```

@*title and optional description*
sequence line(s)
+*optional repeat of title line*
quality line(s)

Figura 3.14: Example of a *fastq* file.

Regarding the pipeline shown in Figure 3.12, the filtering step is important, since errors can occur during the sequencing process. Thus, it is necessary to filter the files received from the sequencers, to assure the quality of the sequences that will be used in the other steps of the pipeline. The filtering step uses softwares such as Prinseq [62], which allows to filter sequences according to the desired quality. Softwares like FastQC [63] can still be used in this step, by receiving *fastq* files as input and generating views of the sequence quality (Figure 3.15).

After the filtering step, we have to group the sequences in order to generate consensus sequences that represent the real biological sequence, what is done in the assembly step. We have two types of assembly: the one with a reference genome; and the *de novo* assembly. In the first one, a genome of the same organism, or of an organism evolutionarily close to the analyzed organism, is used as a guide for the assembling. By using a genome as reference, the assembly can be faster and more precise. But sometimes we do not have an appropriate genome to be used, what can hinder the discovery of sequences being mapped, specific of the organism under study. Figure 3.16 shows an example of assembly with reference genome.

On the other hand, in the *de novo* assembly, the groups are generated by analyzing the overlap of the sequences generated by the sequencer. Only the groups that have enough sequences composing them (groups with good coverage) ensure that the group is reliable.

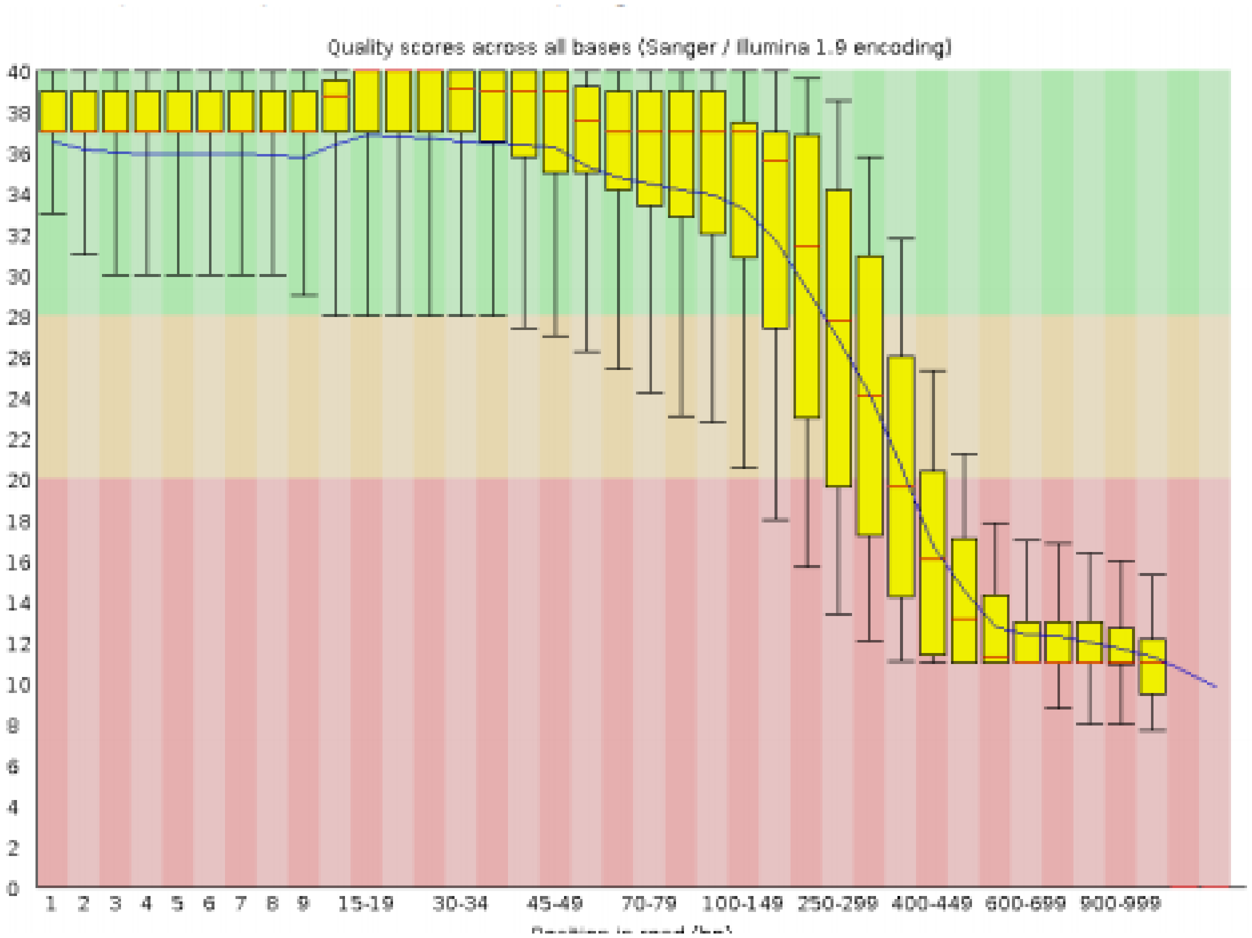


Figura 3.15: Graphic that show the quality of sequences, generated by FastQC [63].

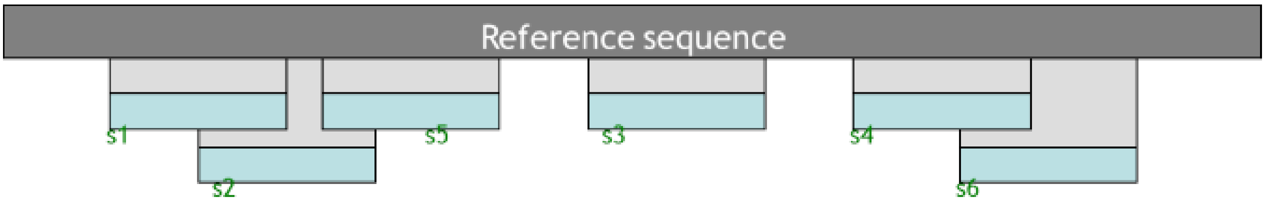


Figura 3.16: Example of assembly using a reference genome, where the reference sequence is an organism evolutionarily close to the analyzed sequence organism, while $s1, s2, s3, s4, s5$ and $s6$ are sequences of the studied organism [64].

As we do not have reference genomes, this assembly process can be slow. Figure 3.17 shows an example of *de novo* assembly.

The last pipeline step, anotation, aims to assign biological functions to the consensus of the sequences grouped in the assembly step. Annotation changes according to the project goal. In transcriptome projects, for example, the annotation aims to describe expressed

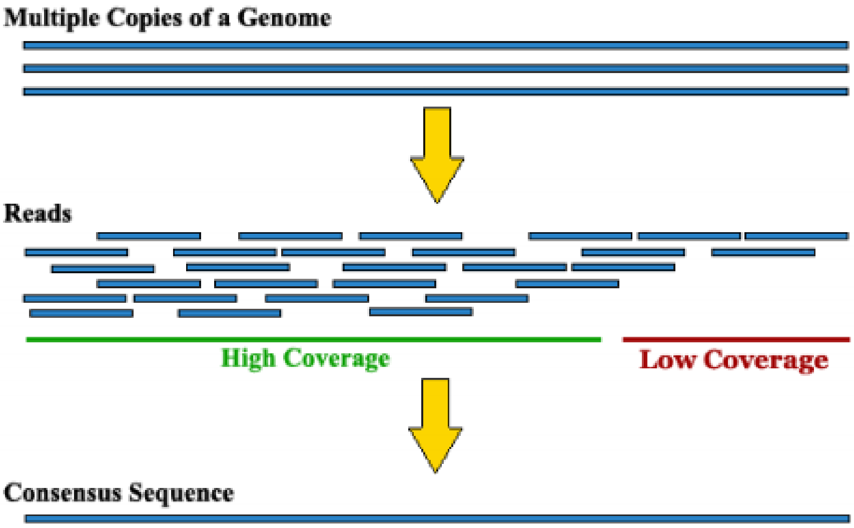


Figura 3.17: Example of a *de novo* assembly, containing areas with high and with low coverage, according to the number of sequences present on the corresponding group [64].

genes and their isoforms, besides their potential roles on the analyzed organism. However, in genome projects, the goal can be the identification of coding genes, and of non-coding genes. To perform annotation, biological databases containing sequences with known biological functions, together with similarity analyzing tools, can be used. One of the most used tools is Basic Local Alignment Search Tool (BLAST) [65], which finds similar regions among sequences, computing local alignments. BLAST finds the function of the sequence by looking for similarities between the sequence under study and each sequence stored in a database, which have know pre-determined functions.

We have many BLAST variations, depending on the studied sequence and the sequences stored in the database:

- blastn, which uses nucleotides as query, and also in the database;
- blastp, which uses amino acids as query, and also in the database;
- blastx, which uses translated nucleotides as query, and amino acids as database;
- tblastn, which uses amino acids as query, and nucleotides translated in amino acids as database;
- tblastx, which uses nucleotides translated in amino acids as query, and also in the database.

In Figure 3.18 we can see how the annotation process works.

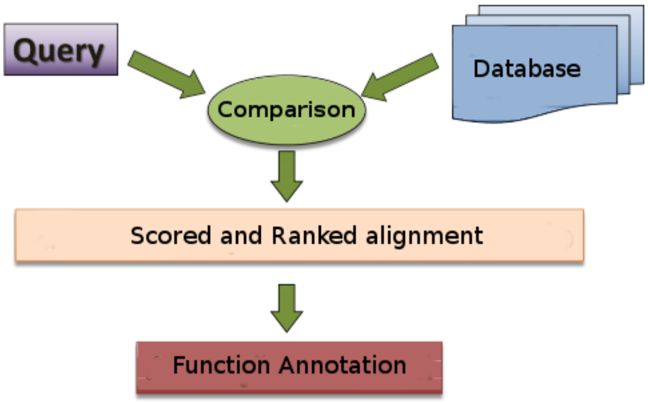


Figura 3.18: General view of the annotation process. A query is the input, that is aligned with sequences in the database, and scored by the BLAST. Similar sequences indicate function conservation.

3.3 Biological aspects of lincRNAs

LncRNAs is usually classified into six major categories: (a) sense or (b) antisense, when the lncRNA overlaps the transcription region of one or more exons of another gene, on the same or the opposite strand, respectively; (c) bidirectional, when the start of the lncRNA transcription and another gene in the opposite strand are close; (d) intronic, when the lncRNAs are derived entirely from introns; (e) enhancer, when the lncRNAs are located in enhancer regions; or (f) intergenic, also called lincRNA, when the lncRNA is located in the interval between two genes [66]. Figure 3.19 illustrates these categories.

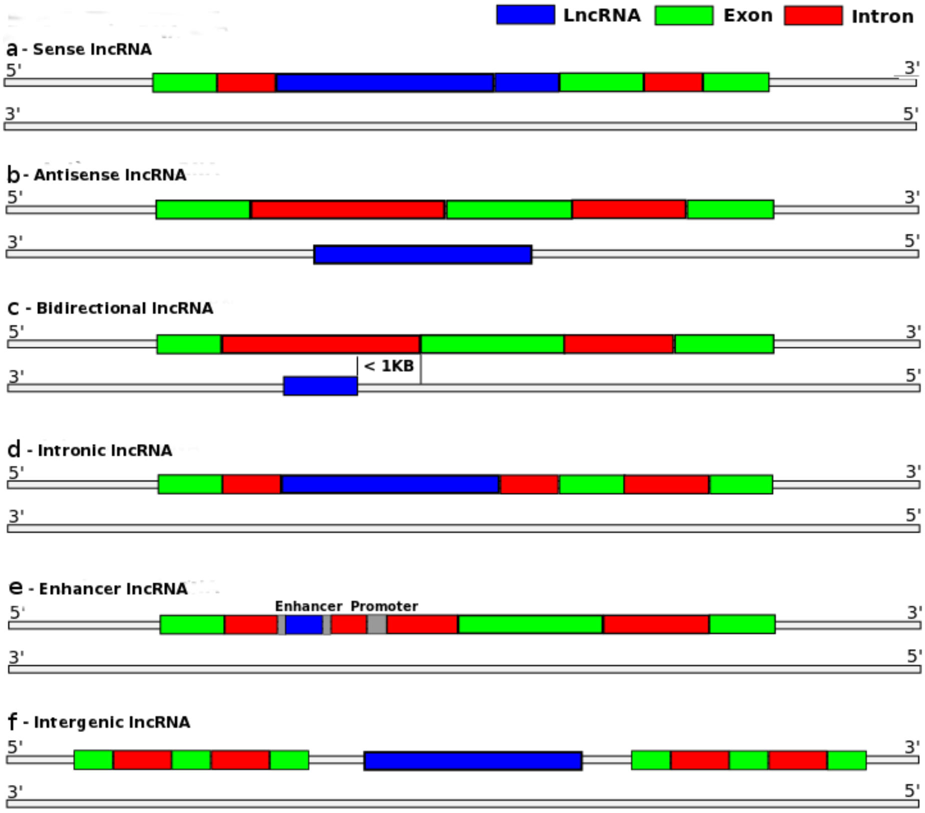


Figura 3.19: LncRNA categories: (a) sense; (b) antisense; (c) bidirectional; (d) intronic; (e) enhancer; and (f) intergenic. Adapted from [66].

Broadly speaking, lncRNAs can be divided in two subsets: lncRNAs that overlap with protein-coding genes; and lincRNAs, found at intergenic regions. The evolutionary history and patterns of conservation (and thereby prediction patterns) of these two lncRNAs subsets are very different. For instance, lncRNAs that overlap with protein-coding genes look like protein-coding genes. They are spliced (predominantly), exhibit elevated conservation (relative to lincRNAs), and are expressed (typically) in a manner that is similar to the protein-coding gene they overlap. Therefore, even with the important roles they play, it is difficult to predict lincRNAs.

The lincRNA classification differs a bit from the other lncRNAs, because they do not have a well defined secondary structure. LincRNAs have been broadly studied due to the fact that they do not overlap any gene [5, 45, 46].

Many lincRNA researches reveal their role in a variety of organisms, performing many different biological roles: Hotair, which may have a role in the chromatin regulation [37]; H19, which may limit the growth of the placenta in mammals [67]; Tincr, the cyran and the megamind, which are necessary for a good embryonic development [68]; HotairM1, which regulates the developmental cycle in maturation of the bone marrow [69]; and Gas5 and Tug1, which can act as a tumor suppressor [38].

LincRNAs are the focus of this project, and although some of their roles are known, e.g, they participate in diseases like cancer [70, 71], there are not broadly used techniques to identify or classify them nor to distinguish lincRNAs from PCTs. Figure 3.20 shows some lincRNAs and their biological roles.

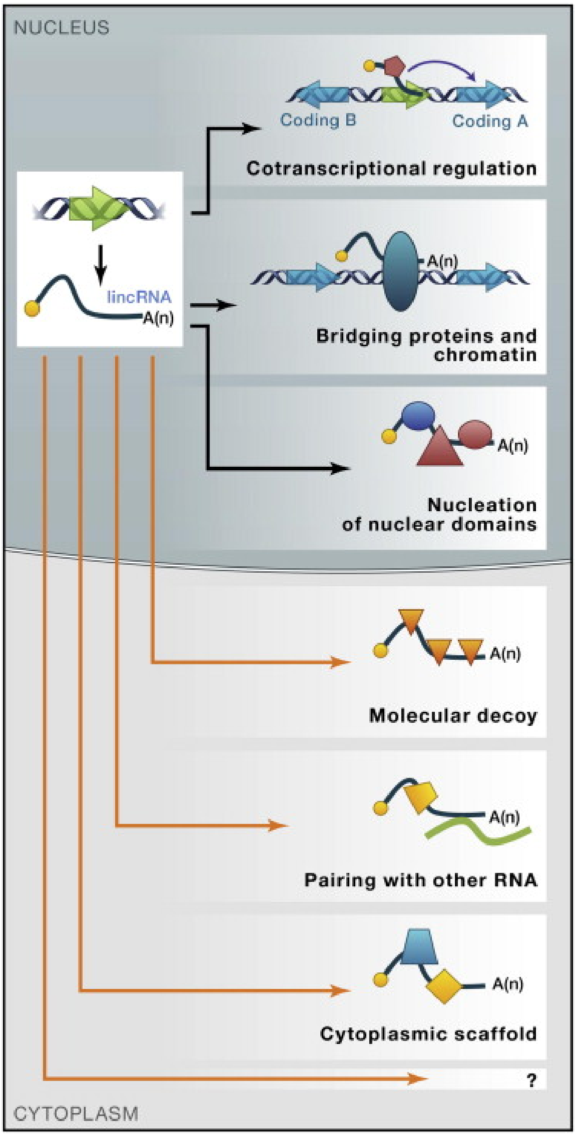


Figura 3.20: Some already discovered biological functions for lincRNAs [5].

Capítulo 4

Machine Learning

In this chapter, we present basic concepts on machine learning, particularly, Support Vector Machine (SVM) and ensemble methods, adopted in this work. In Section 4.1, we present basic concepts of machine learning. In Section 4.2, we discuss the SVM method. In Section 4.3, we introduce the ensemble method. In Section 4.4, we present a literature review with methods that use machine learning algorithms to predict lncRNAs and lincRNAs.

4.1 Basic concepts

Machine learning, in artificial intelligence, focuses on the development of algorithms that detect patterns and learn by experience [72]. In order to achieve this, the main task in machine learning is to build a good model for information extracted from datasets.

In order to build a predictive model, machine learning techniques use feature vectors [73] alongside with a procedure divided in two main steps: training phase and testing phase, both described next.

4.1.1 Training and testing phases

The training phase is the part of the process where the model is generated from the input data [73], and it is where the prediction hypothesis is built.

After building a model on the training phase, this model have to be tested in order to validate its prediction hypothesis, which is done in the so-called testing phase. In this phase, the model's prediction performance can be calculated.

A prediction model can have many goals, among them, clustering data into groups and finding patterns in the data, such that new input data can be classified in these groups or patterns. Prediction models follow some learning paradigms: unsupervised, supervised, learning by reinforcement and semi-supervised, as briefly described.

4.1.2 Learning paradigms

Supervised algorithms are based on the knowledge of the classes being analyzed. Basically, it classifies the input data as belonging to one of the classes, previously known. This classification is done by a function called hypothesis that, according to the features given from a dataset in the training phase, builds a model capable of classifying new input data as belonging to specific classes. Some examples of supervised algorithms are SVM [74], Ctree [75] and KNN [76].

Unlike the previous paradigm, unsupervised learning tries to recognize patterns on a given dataset, in which the labels of the input data are not used. Based on these input data features, the algorithm tries to find patterns so that the input data is labeled and grouped accordingly. The output is composed of sets of input data. Some examples of unsupervised methods are k-medoids [77] and k-means [78].

Learning by reinforcement is a paradigm in which the algorithm learns on each interaction, in order to achieve a final goal. The algorithm interacts with the environment (characterized by elements other than the program itself). A decision made by the program receives a score, used to decide the best classification. The decisions taken by the program receive rewards, which inform the best action to take, given the possible known states of the environment [79]. Some examples of learning by reinforcement are SARSA [80] and LSTD [81].

Finally, we have the semi-supervised learning paradigm, a method that extends the supervised learning by using unsupervised learning techniques. In some cases, its performance overcomes both the unsupervised and supervised learning approaches, if they would be used separately. Usually, the algorithm input dataset is constituted by a group $X = \{x_1, ..., x_{i \in \mathbb{N}}\}$, divided in two groups: (i) $X_l = \{x_1, ..., x_l\}$, in which each x_k has a position in $Y_l = \{y_1, ..., y_l\}$, which corresponds to its class; and (ii) a group $X_u = \{x_1, ..., x_u\}$ of data points with unknown classes [82]. Some examples of semi-supervised learning are Label Spreading [83] and Label Propagation [84].

Given the learning paradigms, the built models, independently from their goals, should have good performance in order to guarantee that their output are reliable. There are distinct ways to measure this performance, and some of them are going to be described next.

4.1.3 Performance measures

Aside from the classification, we have to ensure that the built model is reliable. In order to analyze "how good" is a model, some metrics are used. Most of these metrics are calculated based on the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), from the output classification of the constructed model in the testing phase. Table 4.1 shows the so-called confusion table, often used to visualize the performance of the method.

Tabela 4.1: Confusion table, where we have the number of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN) predicted by the model in the training phase.

Class	Predicted as True	Predicted as False
Input true objects	Number of TP	Number of FN
Input false objects	Number of FP	Number of TN

Using the confusion table we can calculate metrics that evaluate the performance of the model constructed in the training phase. Some metrics will be defined, recall, precision, specificity, F-measure and accuracy, each one measuring a particular aspect of the built model.

Recall shows the rate of TP predicted by the model, and it is calculated by:

$$recall = \frac{TP}{TP + FN}$$

Precision shows the rate of the input data classified as positive, which are really positive, and it is calculated by:

$$precision = \frac{TP}{TP + FP}$$

Specificity calculates the rate of negatives predicted as so, and it is calculated by:

$$specificity = \frac{TN}{TN + FP}$$

F-measure combines precision and recall using a harmonic mean, and it is calculated by:

$$F - measure = \frac{2 * Precision * recall}{Precision + recall}$$

Finally, accuracy is a metric that calculates the general rate of the model:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

The performance can also be represented as graphs. Per Example, the ROC curves can help understanding the ratio of true positives and false positives, by analysing the Area Under the Curve (AUC), which as closest is to the value one as better is the model.

Given the learning paradigms and metrics for the performance measurement, next we describe SVM and ensemble, which are the machine learning algorithms adopted in this work.

4.2 Support Vector Machine

Support Vector Machine (SVM) is a supervised method that classifies groups based on the creation of separation margins. These margins, found by a fraction of the training data, are called support vectors, and they separate sets of data into known labeled classes (Figure 4.1).

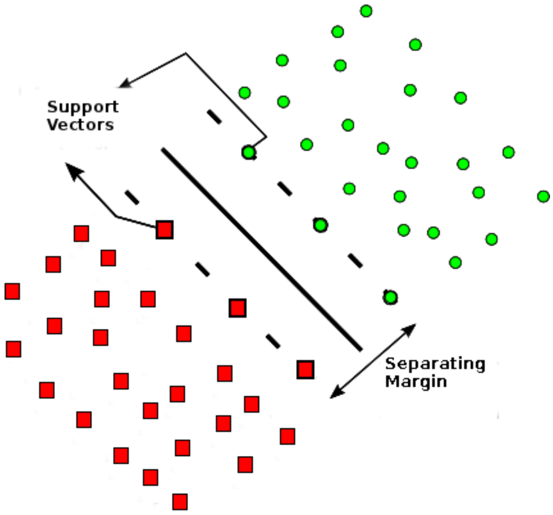


Figura 4.1: Example of support vectors with dimension 2, where the support vectors separate circles from square objects. Adapted from [85].

SVM is a non-parametric method, not limited by the size of the training dataset. Basically, SVM generates models used for classification and regression. In both cases, in order to achieve its tasks, SVM construct hyperplanes in a high dimensional space and select the ones with the largest margin, related to the training data [86]. In the training phase, the classes are separated by a function built by the model generation, called hypothesis. In the testing phase, data is classified according to the model built on the training phase, when the model accuracy can be evaluated.

If SVM tries to find a linear separator in order to divide the dataset in groups, the method can face difficulties using simple linear separation methods for non-linear separable data. In order to solve this problem, SVM uses kernel functions to increase the dimensions of the space, allowing to create linearly separable dataset components in higher dimensions (Figure 4.2).

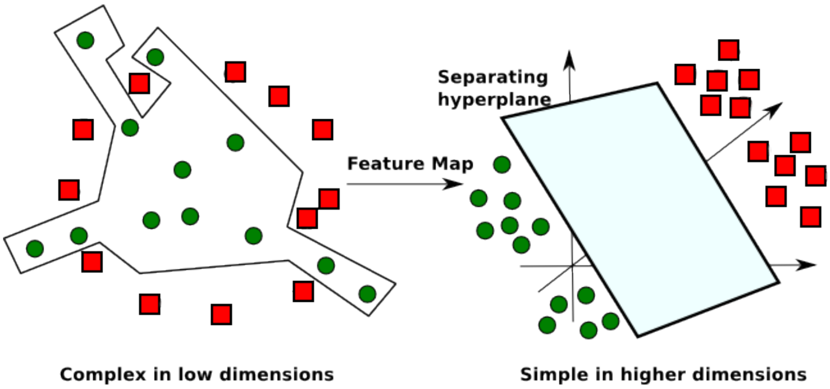


Figura 4.2: Non-linear separable data in low dimension, mapped to a higher dimension, so that the separation of the groups may be simplified in a hyperplane with a higher dimension. Adapted from [87].

One kernel function denotes an inner product in a feature space and it is denoted by $K(x, y) = (\phi(x), \phi(y))$. This feature space has a higher dimension and it is used in such a way that the dataset of the input space transformed into this feature space can be more easily separated. Table 4.2 shows the most commonly used kernel functions.

Tabela 4.2: Most used kernel functions, where $\bullet$ is the internal product, γ and C are constants and X is the input.

Kernel	Fórmula
Linear	$X_i \bullet X_j$
Polynomial	$(\gamma X_i \bullet X_j + C)^d$
RBF (radial)	$exp(-\gamma \mid X_i - X_j \mid^2)$
Sigmoid	$tanh(\gamma X_i \bullet X_j + C)$

In Table 4.2 we can see that some kernel functions, such as the RBF, have a parameter γ. This parameter is adjustable and it has to be used in SVM with an optimal value for better classification results. In addition to the change in γ, we can apply several techniques to try to obtain a SVM model with better accuracy. For example, the change of another parameter, the C (cost), affects the penalty in accepting objects on the wrong side of the margin and can be improved to obtain a better model. An important observation is that the values assigned to C can influence the overfitting problem, because the larger the value of C the more restrict the support vectors are to their respective classes. This means that the model may loose the ability to generalize and may incorrectly sort new data.

As said before, the model performance is an important aspect of the data classification. Some techniques can be used to improve the performance. In this project we use grid search [88] to obtain the optimals C and γ and another technique, called k-fold cross-validation to improve the accuracy of the model, both described next.

In the k-fold cross-validation, data is partitioned into k segments (folds) of the same size. After this division, k training and testing iterations are performed so that, at each iteration, a segment of the data is used as validation while the other $k-1$ segments are used as training. Data is usually stratified when they are partitioned, i.e., they are rearranged to ensure good representativeness for each segment [89]. Figure 4.3 shows an example of the use of k-fold cross-validation, with k = 5.

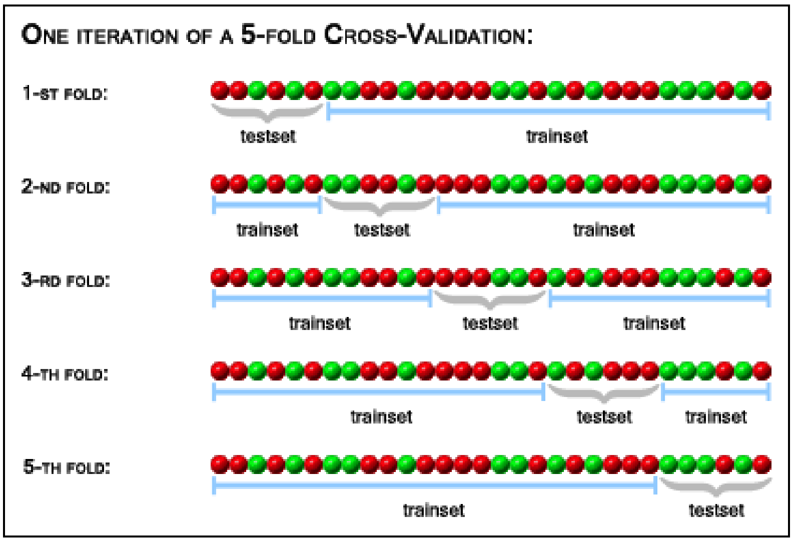


Figura 4.3: Example of k-fold cross-validation with $k = 5$ [6].

In this work, we chose SVM based on: construction of a maximum separating margin, to reduce the classification errors; creation of hyperplans, even if classes are not linearly separable, using kernels; since the method is non-parametric, the capacity of generalization of the constructed model is good. The fact of being non-parametric is one of the main reasons to use SVM, to not limit the quantity of data used in the training phase. This is a good aspect, since whether there are not enough available data labeled as lincRNAs, this does not affect the construction of a good prediction model. Thus SVMs hypotesis space is a big universe when compared to methods that use strictly linear representations and the method can represent complex functions and it is resistant to overfitting [72].

4.3 Ensemble

Ensemble based systems follow the idea of consulting many sources before making a decision, given their known variability and accuracy in other records [90]. This consulting happens because, in most of the time, we can not trust that only knowledge of one source is enough to predict everything the best way possible, given that other sources knowledge can improve this prediction.

Based on this, the machine learning method called Ensemble [73] uses a combination of a set of classifiers estimators, in order to build a classification model with improved generalizability and robustness, when compared to a single estimator model. Ensemble methods are known by its generalization ability, which is, most of the time, better than the results obtained if the used methods are used alone.

According to how the learners are generated, the ensemble methods are divided in two paradigms: sequential ensemble (boosting methods); and parallel ensemble methods (averaging methods) [73]. The averaging methods are based on the construction of several parallel independent estimators, which will have as output a prediction based on the average of their estimators. Normally, their combination is better than a single estimator because the model variance is reduced. On the other side, we have the boosting methods, where several estimators are built sequentially, in order to reduce the bias of these estimators combination.

When using boosting, the combination of several weak models can produce an improved model, meanwhile the averaging methods work better using the combination of strong estimators. At the first part of this work, we proposed a SVM model (described later) to distinguish lincRNAs from PCTs. The model showed an accuracy close to the perfect performance, which made possible to classify the SVM learned used as a strong learner. Given that the SVM was the start point to build the Ensemble, and the fact that parallel ensemble methods tries to exploit the independence between the base learners [73], in this project, a model based on averaging was built.

The base learnes used to build the Ensemble must be complementary in order to improve the model accuracy. In order to achieve this complementarity, our Ensemble used the following supervised classifiers: Suport Vector Machine (SVM), which is a strong method that can take advantage of its independence in an ensemble approach; K-Nearest Neighbor (KNN), which relies on objects similarity to make predictions and can be improved when used in an heterogenios ensemble schema; Conditional Inference Trees (ctree), which can have the prediction error rate reduced when used in boosting or averaging ensemble schemas; and Random Forest(RF), which is an Ensemble of decision trees that can have its prediction error reduced when used with other supervised classifiers in an averaging ensemble. KNN, Ctree and RF will be detailed next.

4.3.1 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is a non parametric lazy learning supervised algorithm, based on a parameter K, which represents the number of neighbors that influences the classification. The distance among the input data generates a classification model. KNN plots the data input in a feature space where we have the notion of distance.

Basically, KNN finds a group of K objects in the training set that are closest to the test input data object, and labels each point as belonging to a particular class in this neighborhood. There are three major parts on this algorithm: a set of labeled input data; a distance metric to compute the distance between two data points; and the value of K, the number of nearest neighbors [91]. Figure 4.4 shows a KNN example.

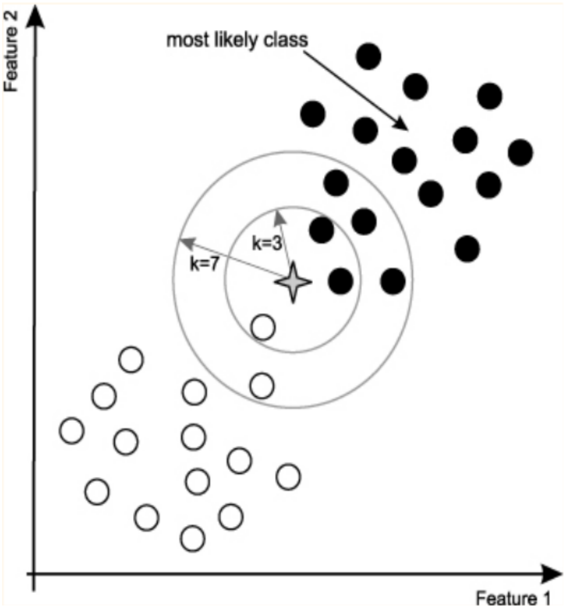


Figura 4.4: Example of KNN with neighbors influence example, for $K = 3$ and $K = 7$ [7].

KNN is very intuitive and it can use different distance metrics, if other knowledge domains are explored. Even with this advantages, KNN has the problem of slow lookups if the number of dimensions is too high. It is important to say that the bigger the training set, the better the KNN efficiency, as it takes advantages of its neighbors and do not generate new insights. KNN was used on this project beacuse of its calculation time speed, its predictive power and its ease to understand output.

4.3.2 Ctrees

Conditional Inference Trees (Ctree) [75] is a non-parametric regression tree estimator that embed tree-structured regression models into a well defined theory of conditional inference procedures. The difference between ctree and other decision trees algorithm, like CART [92] and C4.5 [93], is that it tries to reduce overfitting and a selection bias by generating possible node splits, using a well defined theory of permutation developed by Strasser and Weber [94]. Figure 4.5 shows a ctree example. Ctree was used on this project because of its implicity capacity to perform feature ranking and its simplicity.

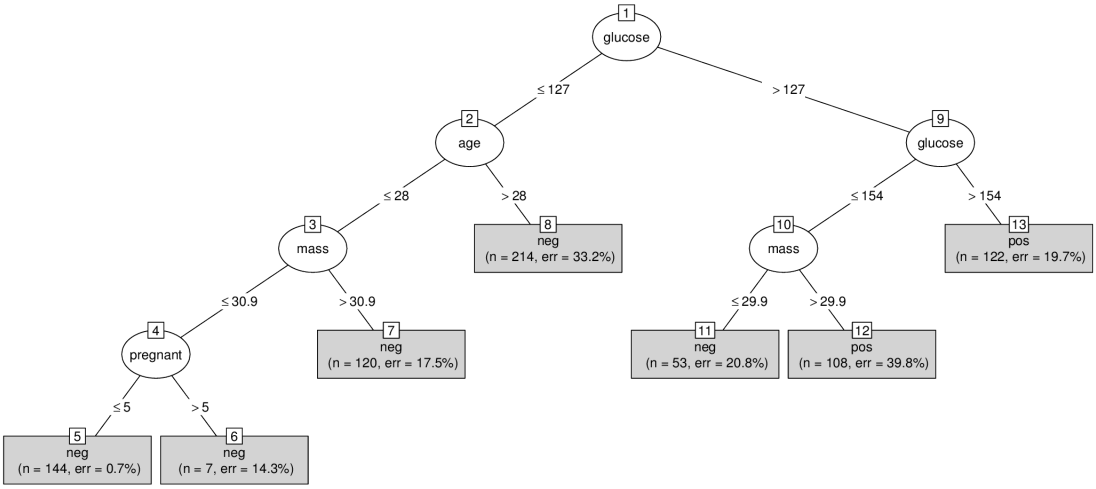


Figura 4.5: Example of a ctree. The higher the node tree the most relevant is the feature in the data classification. In this case when an input data has as feature less or equal 127, it is most likely labeled as negative [95].

4.3.3 Random Forest

Random Forest receives this name because it builds m decision trees as an ensemble, in order to build a better classification model. A random forest is an estimator that fits decision tree classifiers on various subsamples of the dataset, also using the averaging to improve the predictive accuracy at the same time controlling overfitting [96].

In one random forest, each tree in the ensemble is built from a sample in the training set. In addition, when splitting a node, the chosen split is no longer the best split among all the features. Instead, the split that is picked up is the best one among a random part of the features [96]. As a result of this randomness, the bias of the forest usually slightly increases but, due to averaging, its variance also decreases, usually more than compensating for the increase in bias, hence yielding an overall better model. The feature importance can also be extracted by the analysis of the relative rank of a feature used as a decision node in a tree. Features used at the top of the tree are most significant on the final prediction. Figure 4.6 shows a Random Forest example. RF was used on this project because of its implicity capacity to perform feature ranking and its reduced prediction error rate.

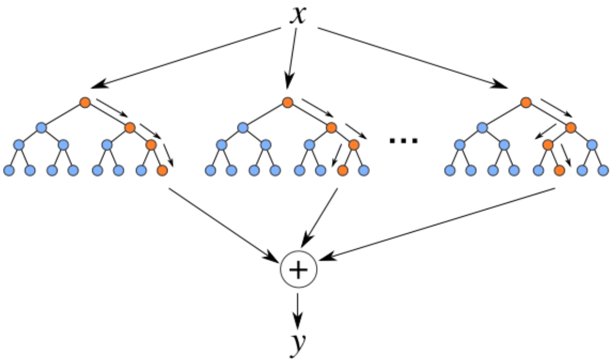


Figura 4.6: Example of Random Forest, where n decision trees are build, given an input x. Note that their n estimators execute an averaging and generate an output y that serves as the ensemble estimator [8].

Given all the machine learning algorithms used on this work, next we present related works that uses machine learning algorithms to discriminate lincRNAs and lncRNAs from PCTs.

4.4 Literature review

4.4.1 Machine learning based tools for distinguishing lncRNAs from PCTs

Han, Siyu, et al. [9] constructed a survey of methods that discriminate PCTs from ncRNAs using machine learning. There are methods based on machine learning algorithms such as: SVM, logistic regression (LR), deep learning (DL) and random forest (RF). In Table 4.3, we list some characteristics of these projects.

As shown in Table 4.3, there are some computational methods, based on machine learning techniques, designed to discriminate ncRNAs from PCTs, and to identify some classes of ncRNAs. Next we briefly discuss some of these methods.

CONC (Coding Or Non-Coding) [97] and CPC (Coding Potential Calculator) [98] have been developed to discriminate protein coding genes from ncRNAs. CONC is slow on analyzing large datasets, CPC works well with known protein coding transcripts but

Tabela 4.3: Methods to discriminate ncRNAs from PCTs [9].

	Year	Testing Datasets	Training Species	Model	Query File Format	Web Interface
CONC [97]	2006	ncRNA	Eukaryotic	SVM	Unknown	Yes
CPC [98]	2007	ncRNA	Eukaryotic	SVM	FASTA	Yes
CNCI [27]	2013	lncRNA	Human and Plant	SVM	FASTA and GTF	No
PLEK [26]	2014	lncRNA	Human and Maize	SVM	FASTA	No
lncRNA-MFDL [29]	2015	lncRNA	Human	DL	Unknown	Unknown
lncRNA-ID [30]	2015	lncRNA	Human and Mouse	RF	BED and FASTA	No
lncRScan-SVM [31]	2015	lncRNA	Human and Mouse	SVM	GTF	No
lncRNApred [32]	2016	lncRNA	Human	RF	FASTA	Web Only
Hugo et. al [33]	2017	lncRNA	Human/Mouse/Zebrafish	SVM/PCA	FASTA	Computer script

may tend to classify novel PCTs into ncRNAs, if they have not been recorded in the protein databases [97].

CNCI [27] , PLEK [26], lncRNA-MFDL [29], lncRNA-ID [30], lncRScan-SVM [31], lncRNApred [32] and Hugo et. al [33] are methods that use machine learning techniques in order to classify lncRNAs. In particular, iSeeRNA [34] and linc-SF [35] use machine learning techniques to classify lincRNAs in human and mouse.

4.4.2 Machine learning based tools to distinguish lincRNAs from PCTs

As we could see on the previous section, we have many methods to identify and classify lncRNAs, but only two methods to distinguish lincRNAs from PCTs, iSeeRNA [34] and linc-SF [35], both described next.

ISeeRNA

ISeeRNA [34] is a SVM based classifier for distinguishing lincRNAs from PCTs. ISeeRNA have a public available webserver and a software for download, which can be used to distinguish lincRNAs in human and mouse assemblies, using *gff* files as input. In order to use SVM, iSeeRNA extracted 10 features to characterize lincRNAs, from three groups: conservation, ORFs, di-nucleotides frequencies and tri-nucleotides frequencies.

SVM was set as binary classifier, with lincRNAs as its positive set and PCTs as the negative set. Optimized SVM parameters C and γ were obtained by using the accompanying grid.py script, with 5,000 randomly selected instances from the training dataset. To obtain the best performance model, 10-fold cross-validation was used. Two models were built, one for human and the other for mouse. The built models presented accuracies of 95.4% for human and 94.2% for mouse.

However, tests with other lincRNAs, in the iSeeRNA web interface, showed many false positives. In other words, they classify many PCTs and other molecules as lincRNAs.

linc-SF

LincRNA classifier, based on Selected Features (linc-SF) [35], was constructed using GA-SVM, using an optimized feature subset. The classifier performance was evaluated to predict lincRNAs, from two independent lincRNA sets composed of human lincRNAs.

In order to build a model using SVM, 74 features were chosen. These features were extracted from three groups: the first one composed of the length of the sequences, frequencies of uni-nucleotides and tri-nucleotides, and the number of ocurrences of G and C on the sequences, forming 70 features in total; the second group was composed of structural features given by RNAfold [99], forming three features; and the third group was formed by the score given by CPC [98].

The method used to build the prediction model was GA-SVM, an algorithm that combines SVM and genetic algorithm (GA). Basically, many rounds using GA were executed, to generate new feature subsets, used with SVM to generate the model.

The method does not present an open source software to evaluate the results, but its recognition rates for the lincRNA human sets achieves 96%. The authors claim that these numbers are good and the method is effective, but such high rates can indicate overfitting.

Capítulo 5

PlantSniffer

Non-coding RNAs (ncRNAs) constitute an important set of transcripts produced in the cells of organisms. Among them, there is a large amount of a particular class of long ncRNAs that are difficult to predict, the so-called long intergenic ncRNAs (lincRNAs), which might play essential roles in gene regulation and other cellular processes. Despite the importance of these lincRNAs, there is still a lack of biological knowledge, and also a few computational methods, specific to organisms, which usually can not be successfully applied to other species, different from those that they have been originally designed to. Besides, prediction of lncRNAs have been performed with machine learning techniques. Particularly, for lincRNA prediction, supervised learning methods have been explored in recent literature. In this context, this work proposes a workflow to predict lincRNAs on plants, considering a pipeline that includes known bioinformatics tools together with machine learning techniques, here Support Vector Machine (SVM). We discuss two case studies that were able to identify novel lincRNAs, in sugarcane (Saccharum spp) and in maize (Zea mays). From the results, we also could identify differentially expressed lincRNAs in sugarcane and maize plants submitted to pathogenic and beneficial microorganisms.

An article with this work was published at *http://www.mdpi.com/2311-553X/3/1/11/htm* (*doi:10.3390/ncrna3010011*).

Capítulo 6

LincSniffer

In this chapter, we propose a method that uses ensemble to distinguish lincRNAs from PCTs. In Section 6.1, we present the method, called LincSniffer. In Section 6.2, we first present results of the method applied to two study cases: *Homo sapiens* (Assembly GRCh38 [100]) and *Mus musculus* (Assembly GRCm38 [101]). Next, we compare the results obtained from our method to other tools found in the literature. In Section 6.3, we present a discussion about LincSniffer and its results.

6.1 Methods

LincSniffer is a method based on a workflow composed of: (1) Data selection and filtering; and (2) Model construction with ensemble. Figure 6.1 describes the method to distinguish lincRNAs from PCTs.

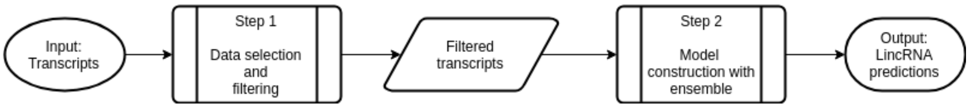


Figura 6.1: LincSniffer workflow: from the input data (lincRNAs and PCTs), step 1 (data selection and filtering) generates the input to build the ensemble model (step 2) to distinguish lincRNAs from PCTs.

6.1.1 Data selection and filtering

Given the difficulties to discriminate some lncRNAs classes from PCTs, and the advantage that lincRNAs have of being found between two genes, the prediction models were constructed using lincRNAs as positive class and PCTs as negative class.

In order to build an accurate dataset, we used transcripts of the HAVANA project [28], which contains manually-curated transcripts. The model performance was tested using two organisms: *Homo sapiens* (Assembly GRCh38 [100]) and *Mus musculus* (Assembly GRCm38 [101]). As PCTs are most known than lincRNAs, the number of PCTs was greater than the lincRNAs, so we chose PCTs annotated as "known"[1] as input data.

Even using data from HAVANA [28] and only "known PCTs", an extra filter step was added to confirm data quality. This confirmation was done by filtering the input data using BLAST [102], as shown in Figure 6.2.

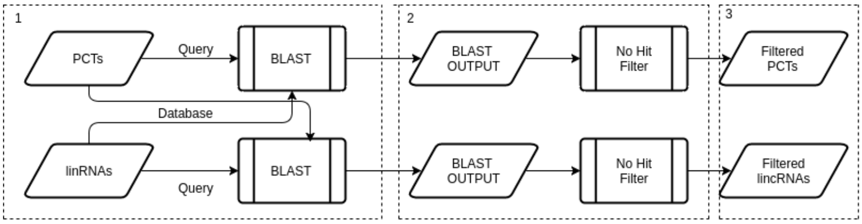


Figura 6.2: Data selection and filtering: 1 - The PCTs and lincRNAs received as input from HAVANA are used as query and database against each other (PCTs X lincRNAs and vice-versa); 2 - The results of BLAST given as output passes through a *no hit* filter script, and only the transcripts not identified in the opposite class are considerated; 3 - The output of the filters guarantees transcripts with high quality.

Even after the filtering process, the number of PCTs was greater than the number of lincRNAs, thus in order to keep the data balance, we fixed the training group size as the amount of lincRNA transcripts available after the filtering step. After fixing this size (s), the positive training data group was defined with size s and the negative was divided in n groups of s. With these groups, n tests were developed, where the positive data group was fixed and prediction performance were calculated by using n different negative groups, in order to validate the prediction score.

6.1.2 Model construction

In order to find the most accurate method for distinguishing lincRNAs from PCTs, individual machine learning methods and their combination were used. The combination of these classifiers, also called ensemble, was used due to its potential capacity to increase the prediction generalizability and performance. According to how the learners are generated, we can divide the ensemble in two paradigms- sequential ensemble (boosting methods) and parallel ensemble methods (averaging methods) [73]. The averaging methods are based on the construction of several parallel independent estimators, which will have as output a prediction based on the average of their estimators. In the boosting approach, several estimators are built sequentially, to decrease the prediction error [96].

[1] "A known gene or transcript matches to a sequence in a public, scientific database such as UniProtKB and NCBI RefSeq" [42].

Usually, ensemble predictions are better then the ones made by single estimators, because the model variance is reduced.

In this project, a model based on averaging of four estimators was built. The averaging method uses a procedure called stacking, where a learner is trained to combine the individual learners (first-level learners) and uses a combiner (meta-learner) to give a prediction [73]. The use of the following supervised classifiers were used on this project: Suport Vector Machine (SVM), K-Nearest Neighbor (KNN), Conditional Inference Trees (Ctree) and Random forest (RF), detailed next.

Feature selection

Training features are a key part of building supervised learning models. They allow to build a correct model, associating specific characteristics to each class, given as input. In our case, the features are biological factors that allow to characterize lincRNAs.

Since lincRNAs are sequences that have ORFs but are not expressed into proteins, i.e., they have poor capacity of coding proteins, we chose features related to ORFs, as follows. The first one is the proportion between the ORF length divided by the sequence length, which captures the percentage of the coding potential capacity. The other feature is the ORF length, explained by Dinger et al. [103], who defined a lincRNA as a transcript with ORF region length less than 100 amino acids. Besides the ORFs, a method based in machine learning for lncRNAs feature selection [104] was used to extract relevant features from the input dataset. As result, 30 features of 2-nucleotides, 3-nucleotides, and 4-nucleotides pattern frequencies more significant to discriminate lincRNAs from PCTs were used (GCGG, TTTT, TCG, AAAA, ACG, TTGT, TAT, TAC,GTT, GTG, AGT, CCGA, TACC, CGTG, CGCT, TACG,TTAG, CGTA, ACCG, CCGT, CGGT, CGAC, CGCA, GCGT,GTAG, CGTT, CGAA, GCGA, CGAT, TAGT). Therefore, we used 32 features in our case studies: ORF proportion, ORF length and 30 nucleotide pattern frequencies.

Single models

A home made script using R [105] was created to build the single models and the ensemble. In order to obtain the best accuracy of each single model in the ensemble, which can affect its accuracy, each single model was built using grid search [88, 106], which search for optimal parameters to build 'the best' model. Figure 6.3 shows how the single prediction models were built.

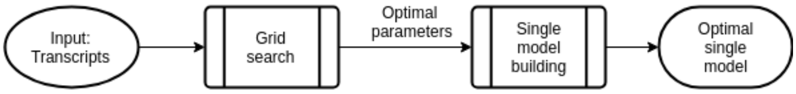


Figura 6.3: Single model construction: the input data received from the data selection and filtering phase, together with the features previously described, are used to build each single model (KNN, Ctree, SVM and RF), which were constructed with parameters optimized by grid search.

Ensemble

With the single models constructed (SVM, KNN, Ctree and RF), an ensemble approach was developed using a stacking model. The stacking built with the parallel execution of the methods (show in Figure 6.4) used two voting strategies (meta-learners) to get the final score: majority voting, where the classification is given when at least three of the single models predict a transcript in the same class; and unanimity voting, where the transcripts are classified as a given class only when all single models say so. In the majority voting in case of a tie, a greater voting weight was attrubuted to the models constructed that presented the better accuracies.

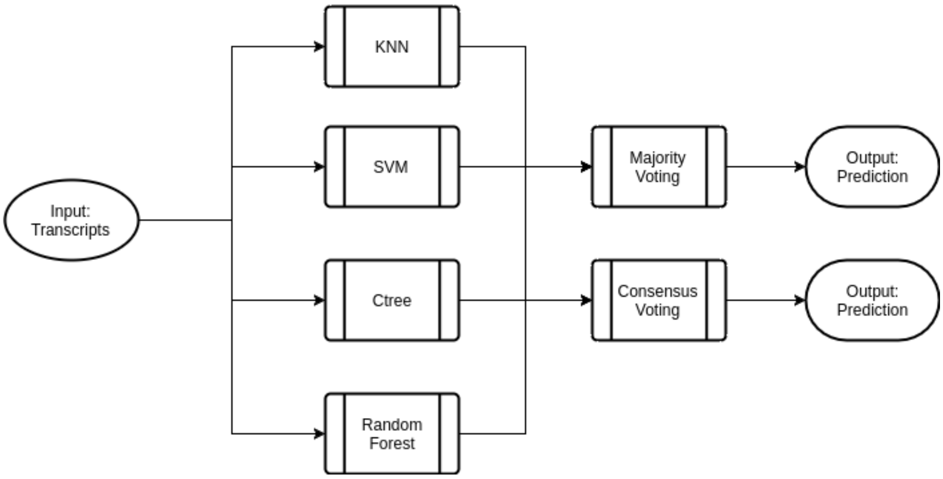


Figura 6.4: ensemble model: the input data received from the filtering phase is used to build four single models (KNN, Ctree, SVM and RF) according to the selected features. Each of the single models gives a prediction of the input. With the prediction of each one, a voting (majority or unanimity voting) is done to get the final score.

6.2 Results

6.2.1 Human

In this section, we discuss the results obtained for the human case, which used data from *Homo sapiens*, assembly GRCh38 [100].

Data and model

As input for the workflow, 13,480 lincRNAs and 40,132 PCTS were used. After the BLAST and transcript size filtering, only 9,094 lincRNAs and 33,695 PCTs remained. In order to keep the balance of the groups and perform a better analysis, these filtered data were divided in one group of 9,094 lincRNAs and 3 groups of 9,094 PCTs. These groups were combined and used separately for the construction of 3 prediction models, where the lincRNA group was used as positive data and each of the 3 groups of PCTs were used as negative data. For each experiment, we had 80% (14,550) transcripts used as training data and 20% (3,636) used as testing data.

First, for each experiment we constructed prediction models by using individual estimators (SVM, Ctree, RF and KNN), and after we constructed ensemble models using two different voting methods: the unanimity and the majority voting. For the unanimity voting, only the transcripts classified as belonging to one class by all the individual estimators were classified as so. On the other hand, the majority voting classifies a transcript as part of one class (positive or negative) according to the individual estimators majority voting.

The lincRNA group was labeled as I and the three PCT groups were labeled as II, III and IV respectively, in order to identify the experiments described next.

Feature ranking

One of the developed experiments is the feature ranking, where each individual estimators used to build the final ensemble prediction model was used to rank the features that were most important to discriminate lincRNAs from PCTs, according to the features used. In this step, we analyzed the ranks given by two of the individual models: Ctree and RF. This is justified since both Ctree and RF are algorithms that uses decision trees to classify their input data. The decision tree visualization can be intuitive for the feature ranking, as they are of easy understanding. Figure 6.5 shows the top nodes of the decision trees constructed with the Ctree algorithm for all groups classification.

As we can in Figure 6.5, the ORF features play a key role to discriminate lincRNAs from PCTs, when using the Ctree algorithm. Not only the ORF features is important to

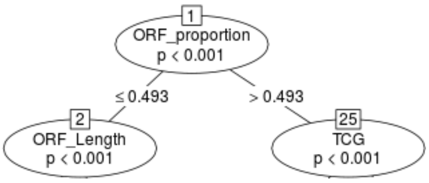


Figura 6.5: Ctree for human data: The top three nodes of the Ctree decision tree for the three groups (I and II, I and III, I and IV).

lincRNAs and PCTs discrimination, but, as we can see in the decision tree, TCG has an important role, which can indicate to possible biological meaning.

In order to validate the feature rank described by the Ctree decision tree, we used the RF algorithm to build a feature ranking list using Gini, as seen in Figure 6.6.

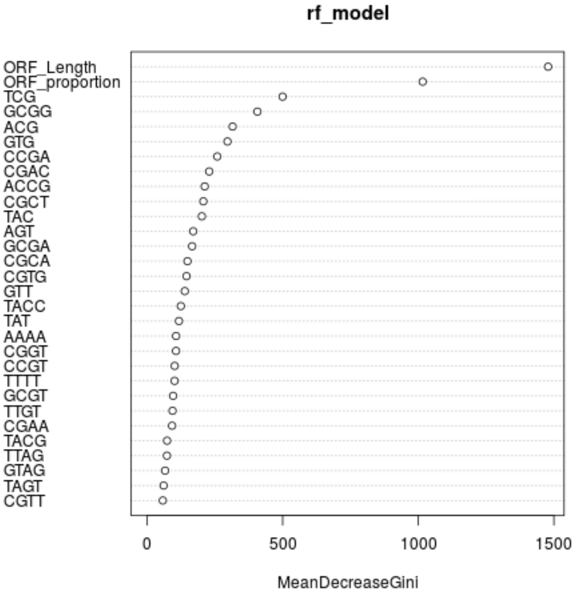


Figura 6.6: RF ranking for human data, the most important features for discriminating lincRNAs from PCTs using the three groups (I and II, I and III and I and IV).

In Figure 6.6 we clearly see that the same features were indicated by Ctree and RF, i.e, ORF proportion, ORF length, GCGG and TCG.

Given this feature ranking, we need to validate the performance of the models, in order to assure its relevance. Next, the evaluation of the single model performance as well as the ensemble models evaluation are described.

Performance evaluation

The performance evaluation allows to confirm if the proposed ensemble prediction model is better than the individual models. Table 6.1 describes the results of the performance

Tabela 6.1: Performance of each single model and the ensemble methods with both voting approaches, for human.

Data Groups	Method	Precision	Recall	F-measure	Accuracy
I and II	Ctree	93%	88%	90%	90%
	KNN	94%	86%	89%	89%
	SVM	89%	89%	90%	89%
	RF	94%	90%	92%	92%
	unanimity	**97%**	**91%**	**94%**	**94%**
	Majority	95%	90%	90%	92%
I and III	Ctree	94%	89%	91%	91%
	KNN	94%	88%	91%	91%
	SVM	89%	88%	89%	89%
	RF	94%	88%	91%	90%
	unanimity	**96%**	**91%**	**93%**	**93%**
	Majority	94%	89%	92%	91%
I and VI	Ctree	92%	88%	90%	90%
	KNN	94%	87%	90%	90%
	SVM	89%	90%	89%	89%
	RF	94%	89%	91%	91%
	unanimity	**97%**	**92%**	**94%**	**94%**
	Majority	95%	90%	93%	92%

evaluation. In this table, it can be seen that the ensemble methods presented a better performance in all the cases, when compared to the single models. The unanimity voting method is the best one, but as it is based on a unanimity voting, it can be more restrict than the majority voting method, what can reduce its generability.

Given these results, Figure 6.7 shows the I and II groups experiment ROC curves, which illustrates a binary classifier (in this case, lincRNA or PCT classes) capacity of correctly cllassify the input. The ROC Curves can help understanding the ratio of true positives and false positives, by analysing the Area Under the Curve (AUC), which as closest is to the value one as better is the model.

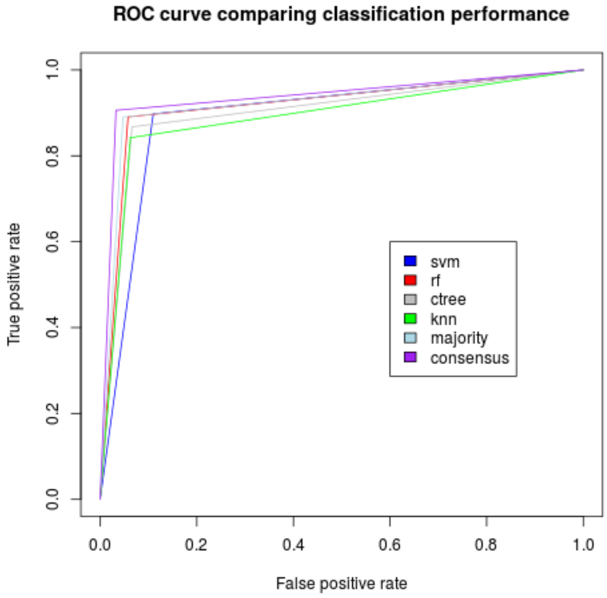


Figura 6.7: Human ROC curves: the ensemble models have good prediction rates when compared to the single models.

6.2.2 Mouse

In this section, we discuss the results obtained for the human case, which used data from *Mus musculus* (Assembly GRCm38 [101]).

Data and model

As input for the workflow, 6,108 lincRNAs and 27,625 PCTS were used. After the BLAST and transcript size filtering, only 4,766 lincRNAs and 25,042 PCTs remained. In order to keep the balance of the groups and perform a better analysis, these filtered data were divided in one group of 4,766 lincRNAs and 5 groups of 4,766 PCTs. These groups were combined and used separately for the construction of 5 prediction models, where the lincRNA group was used as positive data and each of the 5 groups of PCTs were used as negative data. For each experiment, we had 80% (7,622) transcripts used as training data and 20% (1,910) used as testing data.

First, for each experiment we constructed prediction models by using individual estimators (SVM, Ctree, RF and KNN), and after we constructed ensemble models using two different voting methods: the unanimity and the majority voting. For the unanimity voting, only the transcripts classified as belonging to one class by all the individual estimators were classified as so. On the other hand, the majority voting classifies a transcript as part of one class (positive or negative) according to the individual estimators majority voting.

The lincRNA group and the four PCT groups were labeled as I, II, II, IV, V and VI, respectively, in order to identify the experiments described next.

Feature ranking

One of the developed experiments is the feature ranking, where each individual estimators used to build the final ensemble prediction model was used to rank the features that were most important to discriminate lincRNAs from PCTs, according to the features used. In this step, we analyzed the ranks given by two of the individual models: Ctree and RF. This is justified since both Ctree and RF are algorithms that uses decision trees to classify their input data. The decision tree visualization can be intuitive for the feature ranking, as they are of easy understanding. Figure 6.8 shows the top nodes of the decision trees constructed with the Ctree algorithm for all groups classification.

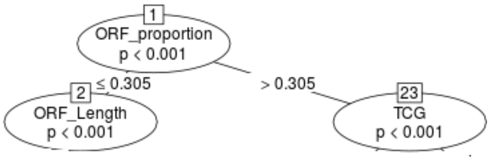


Figura 6.8: Ctree for mouse data: The top three nodes of the Ctree decision tree for the five groups (I and II, I and III, I and IV, I and V and I and VI).

As we can in Figure 6.8, the ORF features play a key role to discriminate lincRNAs from PCTs, when using the Ctree algorithm. Not only the ORF features is important to lincRNAs and PCTs discrimination, but, as we can see in the decision tree TCG has an important role, what can point to possible biological meaning.

In order to validate the feature rank described by the Ctree decision tree, we used the RF algorithm to build a feature ranking list using Gini, as seen in Figure 6.9.

In Figure 6.9 we clearly see that the same features were indicated by Ctree and RF, i.e, ORF proportion, ORF length, GCGG and TCG.

Given this feature ranking, we need to validate the performance of the models, in order to assure its relevance. Next, the evaluation of the single model performance as well as the ensemble models evaluation are described.

Performance evaluation

The performance evaluation allows to confirm if the proposed ensemble prediction model is better than the individual models. Table 6.2 describes the results of the performance evaluation. In this table, it can be seen that the ensemble methods presented a better performance in all the cases, when compared to the single models. The unanimity voting

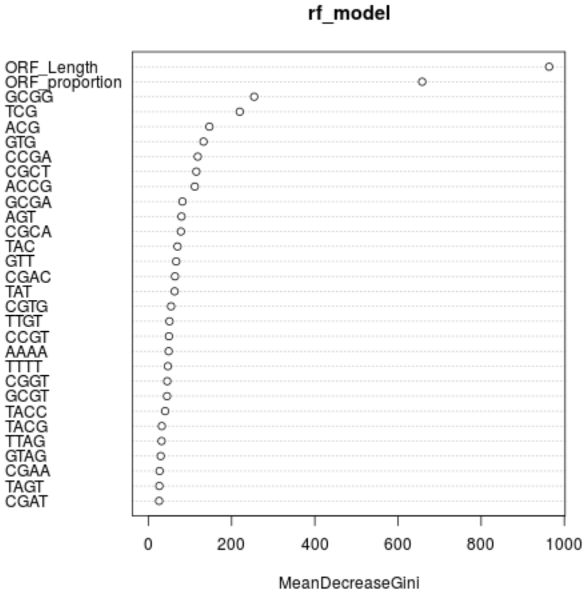


Figura 6.9: RF ranking for mouse data, the most important features for discriminating lincRNAs from PCTs using the five groups (I and II, I and III, I and IV, I and V and I and VI).

method is the best one, but as it is based on a unanimity voting, it can be more restrict than the majority voting method, what can reduce its generability.

Tabela 6.2: Performance of each single model and the ensemble methods with both voting approaches, for mouse.

Data Groups	Method	Precision	Recall	F-measure	Accuracy
I and II	Ctree	92%	91%	90%	91%
	KNN	91%	94%	88%	91%
	SVM	84%	93%	88%	89%
	RF	93%	94%	92%	93%
	unanimity	**96%**	**97%**	**94%**	**96%**
	Majority	94%	93%	**94%**	93%
I and III	Ctree	94%	89%	91%	91%
	KNN	94%	88%	91%	91%
	SVM	86%	92%	89%	89%
	RF	93%	94%	92%	93%
	unanimity	**96%**	**98%**	**93%**	**96%**
	Majority	95%	91%	**93%**	93%
I and VI	Ctree	92%	90%	91%	91%
	KNN	95%	87%	91%	90%
	SVM	86%	92%	89%	89%
	RF	94%	91%	93%	93%
	unanimity	**97%**	**93%**	**95%**	**95%**
	Majority	95%	92%	93%	93%
I and V	Ctree	95%	89%	92%	92%
	KNN	95%	87%	91%	91%
	SVM	85%	90%	88%	88%
	RF	95%	91%	93%	93%
	unanimity	**98%**	**93%**	**95%**	**95%**
	Majority	96%	92%	94%	94%
I and VI	Ctree	92%	91%	91%	91%
	KNN	94%	89%	91%	91%
	SVM	90%	92%	91%	91%
	RF	94%	93%	93%	93%
	unanimity	**97%**	**94%**	**95%**	**95%**
	Majority	95%	93%	94%	94%

Given these results, Figure 6.10 shows the I and II groups experiment ROC curves, which illustrates a binary classifier (in this case, lincRNA or PCT classes) capacity of correctly cllassify the input. The ROC Curves can help understandin the ratio of true positives and false positives, by analysing the Area Under the Curve (AUC), which as closest is to the value one as better is the model.

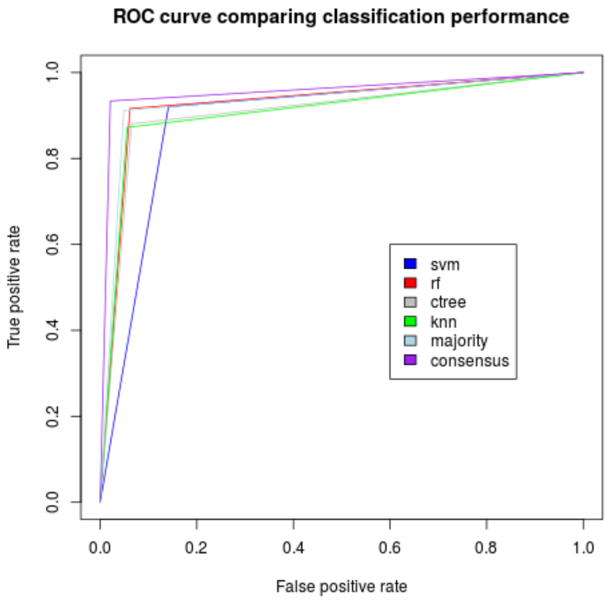


Figura 6.10: Mouse ROC curves: the ensemble models have good prediction rates when compared to the single models.

6.2.3 Methods comparison

We did not compared LincSniffer to linc-SF since it does not have a public available tool. We compared our model to iSeeRNA (*http://137.189.133.71/iSeeRNA/*), the only available tool to discriminate lincRNAs from PCTs found in the literature. ISeeRNA can only distinguish transcripts from mouse (assemblies mm9 or mm10) and human (assembly hg19). ISeeRNA tool only accepts GFF/GTF and $BED12$ input formats, while LincSniffer uses *Fasta* file format. Thus, home made scripts were used to convert $FASTA$ to GFF/GTF, using assembly GRCh37 for human and GRCm38 for mouse.

For *Homo sapiens* (human), we trained a new model with the GRCh37 assembly, since iSeeRNA only accepted this model. Table 6.3 shows the comparison between our method and iSeeRNA. Note that the GRCh37 model was divided in 5 data groups, where I is composed by 1,521 lincRNAs and II, III, VI, V and VI are groups with 1,521 PCTs each. In this table, we can see that iSeeRNA classifies most of the input data as lincRNA. On the other hand, LincSniffer shows balanced results, with an overall performance of 91%, which is a high accuracy when compared to iSeeRNA overall accuracy of 56%.

Tabela 6.3: Comparison of LincSniffer and iSeeRNA, for human.

Data Groups		iSeeRNA			LincSniffer	
	lincRNAs	PCTs	Accuracy	lincRNAs	PCTs	Accuracy
I and II	1466/1521	239/1521	56%	1422/1521	1354/1521	91%
I and III	1466/1521	218/1521	55%	1426/1521	1348/1521	91%
I and VI	1466/1521	233/1521	55%	1420/1521	1332/1521	91%
I and V	1466/1521	219/1521	55%	1430/1521	1348/1521	91%
I and VI	1466/1521	225/1521	55%	1420/1521	1347/1521	90%

Regarding *Mus musculus* (mouse), we compared LincSniffer directly to iSeeRNA since both models use the same assembly (GRCm38). Table 6.4 shows these comparisons. Similarly to the human case, iSeeRNA false positive rate is high, causing a decrease in its accuracy, which means that iSeeRNA classifies most of the input data as lincRNA, while LincSniffer show balanced results, with an overall performance of 90%. We note that LincSniffer 90% accuracy of this experiment is different from the 96% accuracy reported in Section 6.2.2. Here, lincRNAs or PCTs are classified as so, only if all the four methods agree.

Tabela 6.4: Comparison of LincSniffer and iSeeRNA, for mouse.

Data Groups		iSeeRNA			LincSniffer	
	lincRNAs	PCTs	Accuracy	lincRNAs	PCTs	Accuracy
I and II	931/955	217/955	60.10%	871/955	854/955	90%
I and III	931/955	188/955	58.58%	873/955	856/955	90%
I and IV	931/955	195/955	58.95%	873/955	852/955	90%
I and V	931/955	193/955	58.84%	882/955	840/955	90%
I and VI	931/955	214/955	59.94%	867/955	865/955	90%

6.3 Discussion

In this work, we proposed an ensemble model to discriminate lincRNAs from PCTs. In order to test this model, we performed two experiments with *Homo sapiens* (human), assembly GRCh38, and *Mus musculus* (mouse), assembly GRCm38.

In general the ensemble models (majority voting and unanimity voting) presented better accuracies when compared to the single models. It is important to note that the unanimity voting can lead to overfitting.

Comparing the results related at iSeeRNA [34] and linc-SF [35] for *H. sapiens*, 96.1% and 96.19%, respectively, LincSniffer showed an accuracy of 94%, using the unanimity voting. The best approach would be to compare the tools by runing them with the same datasets. But, linc-SF does not have a public tool and iSeeRNA only works with a specific assembly. The results obtained when running iSeeRNA with our dataset showed a different performance from the ones presented in their article. ISeeRNA presented a best accuracy of 56%, while LincSniffer accuracy was 91%, having shown high false positive prediction rates.

Regarding *M. musculus*, our model showed an accuracy of 96%, using the unanimity voting approach, against the 94.7% reported in iSeeRNA. As linc-SF works only with human, we could not make comparisons with it. The results obtained when running iSeeRNA with our dataset showed a different performance from the ones presented in their article. ISeeRNA presented a best accuracy of 60.10%, while LincSniffer accuracy was 90%, having shown high false positive prediction rates.

LincSniffer was designed to distinguish lincRNAs from PCTs, having shown a good performance according to the tests and comparisons made. Thus, LincSniffer generalize better than iSeeRNA. Also, our tests indicated that ORF length and ORF proportion are important for lincRNA and PCT discrimination, which was confirmed by experiments [107, 108] that indicate the ORF lengths of lincRNAs between 50 and 100 amino acids. Besides the ORFs, our tests indicated that the number of TCG occurrencies in the transcripts had a key role, which has to be biologicaly confirmed.

LincSniffer scripts, togheter with all the tests and results are public available at *https://github.com/lmaciel vieira/LincSniffer*.

Next steps include improving LincSniffer usability and parallelizing its execution. Also, we plan to include tests with some other ensemble meta-learners, such as arithmetic mean, geometric mean and logistic regression.

Capítulo 7

Conclusion

In this work, we built two models based on machine learning to discriminate lincRNAs from PCTs. The first one was a pipeline, using SVM models, to discriminate lincRNAs from PCTs, in plants. We developed two case studies for sugarcane and maize. Regarding sugarcane, we found putative 67 lincRNAs, being 1 of them tested in the laboratory. Besides, lincRNAs differentially expressed were investigated, in libraries treated with *Acidovorax avenae spp avenae*, the causal agent of the red stripe disease, and two control libraries. In total, 46 of the 67 predicted lincRNAs were differentially expressed, when comparing the sugarcanes with red stripe disease with the control ones. Regarding maize, we worked with transcripts obtained from Illumina HiSeq, noting that eight libraries were produced, two treated with *Herbaspirillum seropedicae* and *Azospirillum brasilense*, respectively, while the other two were controls. In this case, our SVM model exhibited an accuracy of 99%. Also in this case, differentially expressed lincRNAs were investigated, comparing the treated libraries with the control ones.

The second model was based on ensemble, to discriminate lincRNAs from PCTs in human and mouse. For *Homo sapiens*, comparing the results related at iSeeRNA [34] and linc-SF [35], 96.1% and 96.19%, respectively, LincSniffer showed an accuracy of 94%. For *Homo sapiens*, the results obtained when running iSeeRNA with our dataset showed a different performance from the ones presented in their article. ISeeRNA presented a best accuracy of 56%, while LincSniffer accuracy was 91%, having shown high false positive prediction rates. Regarding *Mus musculus*, our model showed an accuracy of 96%, against the 94.7% reported in iSeeRNA. For *Mus musculus*, the results obtained when running iSeeRNA with our dataset showed a different performance from the ones presented in their article. ISeeRNA presented a best accuracy of 60.10%, while LincSniffer accuracy was 90%, having shown high false positive prediction rates.

7.1 Contributions

Both methods proposed in this work show that computational methods based on machine learning to distinguish lincRNAs from PCTs are useful to indicate potential lincRNAs, which can be experimentally validated further. LincRNAs show different characteristics depending on the species, and probably the biological characteristics have to investigated in each case.

PlantSniffer method was published (see Vieira et al. [36]), having been cited in Mokhtarzad et al. [109] and Thiebaut et al. [110]. LincSniffer is public available at https://github.com/lmacielvieira/LincSniffer.

7.2 Future work

For PlantSniffer, we plan to develop an ensemble based model, and perform more case studies with data available in the literature. Another interesting work is to use genomic data for the plants that have been sequenced, to improve predictions of PlantSniffer.

For LincSniffer, the next step is to develop more case studies with lincRNAs found in public databases. We also intend to use different meta-estimators and a boosting approach, as well as more refined models of individual learners with complementary behaviours, to improve the ensemble method.

Referências

[1] T. Shafee and R. Lowe. Eukaryotic and prokaryotic gene structure. *Wiki J Med*, 4(1):002, 2017. x, 13

[2] Splicing. https://pt.wikipedia.org/wiki/Splicing. Accessed: 2015-11-15. x, 14

[3] Translation – process of polymerization of amino acids to form a polypeptide. http://www.biologydiscussion.com/biology/translation-process-of-polymerization-of-amino-acids-to-form-a-polypeptide/1888. Accessed: 2016-01-23. x, 15

[4] M. Carvalho, D. Silva, et al. Sequenciamento de DNA de nova geração e suas aplicações na genômica de plantas. *Ciência Rural*, 40(3):735–744, 2010. x, 17, 18

[5] I. Ulitsky and D. Bartel. LincRNAs: genomics, evolution, and mechanisms. *Cell*, 154(1):26–46, 2013. xi, 1, 5, 25, 26

[6] Cross-validation. http://stats.stackexchange.com/questions/1826/cross-validation-in-plain-english. Accessed: 2016-01-06. xi, 32

[7] KNN classifier approach. https://www.researchgate.net/figure/297728234_fig3_Figure-7k-NN-classifier-approach. Accessed: 2017-11-12. xi, 34

[8] Decision Tree e Random Forest. http://carlosbaia.com/2016/12/24/decision-tree-e-random-forest/. Accessed: 2017-10-10. xi, 36

[9] S. Han, Y. Liang, Y. Li, and W. Du. Long non-coding RNA identification: comparing machine learning based tools for long non-coding transcripts discrimination. xiii, 36, 37

[10] J. Watson and F. Crick. Molecular structure of nucleic acids. *Nature*, 171(4356):737–738, 1953. 1, 5

[11] J. Setubal, J. e Meidanis. *Introduction to Computational Molecular Biology*. PWS Pub., 1997. 1, 5, 9, 12, 16

[12] P. Alvarez. Pipelines para transcritomas obtidos por sequenciadores de alto desempenho. *Monografia de Graduação. Departamento de Ciência da Computação. Universidade de Brasília*, 2009. 5

[13] J. Wu, D. Delneri, R. O'Keefe, et al. Non-coding RNAs in Saccharomyces cerevisiae: what is the function? *Biochemical Society Transactions*, 40(4):907, 2012. 1, 5

[14] S. Eddy. Non-coding RNA genes and the modern RNA world. *Nature Reviews Genetics*, 2(12):919–929, 2001. 1, 5

[15] C. Ponting, P. Oliver, and W. Reik. Evolution and functions of long noncoding RNAs. *Cell*, 136(4):629–641, February 2009. 1, 5

[16] U. Ørom and R. Shiekhattar. Noncoding RNAs and enhancers: complications of a long-distance relationship. *Trends in Genetics*, 27(10):433–439, 2011. 1, 5

[17] H. Timmers and L. Tora. The spectacular landscape of chromatin and ncRNAs under the tico sunlight. *EMBO Reports*, 11(3):147–149, 2010. 5

[18] M. Szcześniak, W. Rosikiewicz, and I. Makałowska. Cantatadb: A collection of plant long non-coding RNAs. *Plant and Cell Physiology*, 57(1):e8–e8, 2016. 6

[19] G. Eades, B. Wolfson, Y. Zhang, Q. Li, Y. Yao, and Q. Zhou. lincrna-ror and mir-145 regulate invasion in triple-negative breast cancer via targeting arf6. *Molecular Cancer Research*, 13(2):330–338, 2015. 6

[20] Y. Chen, G. Wei, H. Xia, H. Yu, Q. Tang, and F. Bi. Down regulation of lincrna-p21 contributes to gastric cancer development through hippo-independent activation of yap. *Oncotarget*, 8(38):63813, 2017. 6

[21] A. Bernal, U. Ear, and N. Kyrpides. Genomes OnLine Database (GOLD): a monitor of genome projects world-wide. *Nucleic Acids Research*, 29(1):126–127, 2001. 2, 6

[22] L. Sabin, M. Delás, and G. Hannon. Dogma derailed: The many influences of RNA on the genome. *Molecular Cell*, 49(5):783–794, 2013. 2, 6

[23] H. Wang, Q.-W. Niu, H.-W. Wu, J. Liu, J. Ye, N. Yu, and N.-H. Chua. Analysis of non-coding transcriptome in rice and maize uncovers roles of conserved lncRNAs associated with agriculture traits. *The Plant Journal*, 84(2):404–416, 2015. 2, 6

[24] L. Li, S. R Eichten, K. Shimizu, R.and Petsch, C.-T. Yeh, W. Wu, A. M Chettoor, S. A Givan, R. A Cole, J. E. Fowler, et al. Genome-wide discovery and characterization of maize long non-coding RNAs. *Genome Biology*, 15(2):R40, 2014. 2, 6

[25] Y.-C. Zhang, J.-Y. Liao, Z.-Y. Li, Y. Yu, J.-P. Zhang, Q.-F. Li, L.-H. Qu, W.-S. Shu, and Y.-Q. Chen. Genome-wide screening and functional analysis identify a large number of long noncoding RNAs involved in the sexual reproduction of rice. *Genome Biology*, 15(12):512, 2014. 2, 6

[26] A. Li, J. Zhang, and Z. Zhou. PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme. *BMC Bioinformatics*, 15(1):1, 2014. 2, 6, 37

[27] L. Sun, H. Luo, D. Bu, G. Zhao, K. Yu, C. Zhang, Y. Liu, R. Chen, and Y. Zhao. Utilizing sequence intrinsic composition to classify protein-coding and long non-coding transcripts. *Nucleic Acids Research*, page gkt646, 2013. 2, 6, 37

[28] L. G Wilming, J. GR Gilbert, K. Howe, S Trevanion, T Hubbard, and Jennifer L Harrow. The vertebrate genome annotation (vega) database. *Nucleic Acids Research*, 36(suppl 1):D753–D760, 2008. 2, 40, 41

[29] X. Fan and S. Zhang. lncRNA-MFDL: identification of human long non-coding RNAs by fusing multiple features and using deep learning. *Molecular Biosystems*, 11(3):892–897, 2015. 2, 37

[30] R. Achawanantakun, J. Chen, Y. Sun, and Y. Zhang. LncRNA-ID: Long non-coding RNA identification using balanced random forests. *Bioinformatics*, 31(24):3897–3905, 2015. 2, 37

[31] L. Sun, H. Liu, L. Zhang, and J. Meng. lncRScan-SVM: A tool for predicting long non-coding RNAs using support vector machine. *PloS one*, 10(10):e0139654, 2015. 2, 37

[32] C. Pian, G. Zhang, Z. Chen, Y. Chen, J. Zhang, T. Yang, and L. Zhang. LncRNA-pred: Classification of Long Non-coding RNAs and protein-coding transcripts by the ensemble algorithm with a new hybrid feature. *PloS one*, 11(5):e0154567, 2016. 2, 37

[33] H. W Schneider, T. Raiol, M. M Brigido, M. E. MT Walter, and P. F Stadler. A support vector machine based method to distinguish long non-coding rnas from protein coding transcripts. *BMC genomics*, 18(1):804, 2017. 2, 37

[34] K. Sun, X. Chen, P. Jiang, X. Song, H. Wang, and H. Sun. ISeeRNA: identification of long intergenic non-coding RNA transcripts from transcriptome sequencing data. *BMC Genomics*, 14 Suppl 2:S7, 2013. 2, 6, 37, 52, 54

[35] Y. Wang, Y. Li, Q. Wang, Y. Lv, S. Wang, X. Chen, X. Yu, W. Jiang, and X. Li. Computational identification of human long intergenic non-coding RNAs using a GA-SVM algorithm. *Gene*, 533(1):94–99, 2014. 2, 37, 38, 52, 54

[36] L. Vieira, C. Grativol, F. Thiebaut, T. Carvalho, P. Hardoim, A. Hemerly, S. Lifschitz, P. C. Ferreira, and M. E. MT Walter. PlantRNA_sniffer: A svm-based workflow to predict long intergenic non-coding RNAs in plants. *Non-Coding RNA*, 3(1):11, 2017. 3, 55

[37] T. Mercer, M. Dinger, and J. Mattick. Long non-coding RNAs: insights into functions. *Nature Reviews Genetics*, 10(3):155–159, 2009. 6, 25

[38] J. Li, M. Zhang, G. An, and Q. Ma. LncRNA TUG1 acts as a tumor suppressor in human glioma by promoting cell apoptosis. *Experimental Biology and Medicine*, 2016. 6, 26

[39] C. Tong, Q. Chen, L. Zhao, J. Ma, E. Ibeagha-Awemu, and X. Zhao. Identification and characterization of long intergenic noncoding RNAs in bovine mammary glands. *BMC Genomics*, 18(1):468, 2017. 6

[40] L. Wang, X. Ma, X. Xu, and Y. Zhang. Systematic identification and characterization of cardiac long intergenic noncoding RNAs in zebrafish. *Scientific Reports*, 7(1):1250, 2017. 6

[41] K. Etebari, S. Asad, G. Zhang, and S. Asgari. Identification of Aedes aegypti long intergenic non-coding RNAs and their association with wolbachia and dengue virus infection. *PLoS neglected tropical diseases*, 10(10):e0005069, 2016. 6

[42] Ensembl. `http://www.ensembl.org/index.html`. Accessed: 2016-01-21. 6, 41

[43] Havana. `http://vega.sanger.ac.uk/index.html`. Accessed: 2016-01-21. 6

[44] lncRNADisease. `http://www.cuilab.cn/lncrnadisease`. Accessed: 2015-11-15. 6

[45] M. Sauvageau, L. Goff, S. Lodato, B. Bonev, A. Groff, C. Gerhardinger, D. Sanchez-Gomez, E. Hacisuleyman, E. Li, M. Spence, et al. Multiple knockout mouse models reveal lincRNAs are required for life and brain development. *Elife*, 2:e01749, 2013. 7, 25

[46] P. Shuai, D. Liang, S. Tang, Z. Zhang, C.-Y. Ye, Y. Su, X. Xia, and W. Yin. Genome-wide identification and functional prediction of novel and drought-responsive lincRNAs in populus trichocarpa. *Journal of Experimental Botany*, page eru256, 2014. 7, 25

[47] Composição dos nucleotídeos. `http://geneticavirtual.webnode.com.br`. Accessed: 2016-01-26. 10

[48] T. Silva. SOM-Portrait: um método para identificar RNAs não codificadores utilizando Mapas Auto-Organizáveis. monografia. Departamento de Ciência da Computação. Universidade de Brasília. 2009. 9

[49] RNA. `http://biology.about.com/od/molecularbiology/ss/rna.htm`. Accessed: 2016-01-23. 10

[50] T. Silva. Identificação de RNA não-codificador utilizando redes neurais artificiais de treinamento não supervisionado. Dissertação de mestrado. Departamento de Ciência da Computação. Universidade de Brasília. 2012. 10

[51] Five questions for David Root: RNA Interference explained. `https://www.broadinstitute.org/blog/five-questions-david-root-rna-interference-explained`. Accessed: 2016-01-23. 11

[52] A. Machado-Lima, H. Del Portillo, and A. Durham. Computational methods in noncoding RNA research. *Journal of Mathematical Biology*, 56(1-2):15–49, 2008. 10

[53] Central dogma of molecular biology. http://studyfaq.com/blog/central-dogma-of-molecular-biology. Accessed: 2016-12-05. 11

[54] P. Clote and R. Backofen. *Computational molecular biology: an introduction a self contained approach to bioinformatics*. Chichester Wiley, 2000. 12

[55] The warak warak method. https://biologywarakwarak.wordpress.com. Accessed: 2015-11-15. 15

[56] The genetic code. https://www.khanacademy.org/science/biology/gene-expression-central-dogma/central-dogma-transcription/a/the-genetic-code-discovery-and-properties. Accessed:2018-02-05. 16

[57] S. Bennett. Solexa ltd. *Pharmacogenomics*, 5(4):433–438, 2004. 17

[58] G. Porreca. Genome sequencing on nanoballs. *Nature Biotechnology*, 28(1):43–44, 2010. 18

[59] J. Thompson and K. Steinmann. Single molecule sequencing with a heliscope genetic analysis system. *Current Protocols in Molecular Biology*, pages 7–10, 2010. 18

[60] Z. Wang, M. Gerstein, and M. Snyder. RNA-seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009. 19

[61] T. Attwood, A. Gisel, E. Bongcam-Rudloff, and N. Eriksson. *Concepts, historical milestones and the central place of bioinformatics in modern biology: a European perspective*. INTECH Open Access Publisher, 2011. 20

[62] R. Schmieder and R. Edwards. Quality control and preprocessing of metagenomic datasets. *Bioinformatics*, 27(6):863–864, 2011. 21

[63] FastQC: A quality control application for FastQ data. http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc. Accessed: 2016-01-23. 21, 22

[64] Montagem. http://compbio.davidson.edu/phast/. Accessed: 2016-01-26. 22, 23

[65] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. 23

[66] Y. Devaux, J. Zangrando, B. Schroen, E. Creemers, T. Pedrazzini, C. Chang, G. Dorn II, T. Thum, and S. et al. Heymans. Long noncoding RNAs in cardiac development and ageing. *Nature Reviews Cardiology*, 12(7):415–425, 2015. 24, 25

[67] A. Keniry, D. Oxley, P. Monnier, M. Kyba, L. Dandolo, G. Smits, and W. Reik. The h19 lincRNA is a developmental reservoir of mir-675 that suppresses growth and Igf1r. *Nature Cell Biology*, 14(7):659–665, 2012. 25

[68] I. Ulitsky, A. Shkumatava, C. Jan, H. Sive, and D. Bartel. Conserved function of lincRNAs in vertebrate embryonic development despite rapid sequence evolution. *Cell*, 147(7):1537–1550, 2011. 25

[69] X. Zhang, S. Weissman, and P. Newburger. Long intergenic non-coding RNA hotairm1 regulates cell cycle progression during myeloid maturation in nb4 human promyelocytic leukemia cells. *RNA Biology*, 11(6):777–787, 2014. 25

[70] J.-H. He, Z.-P. Han, and Y.-G. Li. Association between long non-coding RNA and human rare diseases (review). *Biomedical Reports*, 2(1):19–23, 2014. 26

[71] M. Huarte and J. Rinn. Large non-coding RNAs: missing links in cancer? *Human Molecular Genetics*, 19(R2):R152–R161, 2010. 26

[72] S. Russell and P. Norvig. *AI a modern approach*, volume 3. Pearson, 2010. 27, 32

[73] Z.-H. Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012. 27, 33, 41, 42

[74] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 28

[75] T. Hothorn, K. Hornik, and A. Zeileis. Unbiased recursive partitioning: A conditional inference framework. *Journal of Computational and Graphical statistics*, 15(3):651–674, 2006. 28, 35

[76] N. S. Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992. 28

[77] L Kaufman and PJ Rousseeuw. Clustering by means of medoids [w:] statistical data analysis based on the ll-norm and related methods, red. y. dodge, 1987. 28

[78] J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967. 28

[79] R. Sutton and A. Barto. Reinforcement learning: an introduction. The MIT Press. *Cambridge, MA*, 1998. 28

[80] G. A Rummery and M. Niranjan. *On-line Q-learning using connectionist systems*, volume 37. University of Cambridge, Department of Engineering, 1994. 28

[81] J. A Boyan. Technical update: Least-squares temporal difference learning. *Machine Learning*, 49(2):233–246, 2002. 28

[82] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006. 28

[83] Z. Xiaojin and G. Zoubin. Learning from labeled and unlabeled data with label propagation. *Technical Report CMU-CALD-02–107, Carnegie Mellon University*, 2002. 28

[84] D. Zhou, O. Bousquet, T. N Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In *Advances in neural information processing systems*, pages 321–328, 2004. 28

[85] Big data optimization at SAS. http://www.maths.ed.ac.uk/~prichtar/Optimization_and_Big_Data/slides/Polik.pdf. Accessed: 2016-01-06. 30

[86] S. Haykin. A comprehensive foundation. *Neural Networks*, 2(2004), 2004. 30

[87] SVM- Support Vector Machines. https://www.dtreg.com/solution/view/20. Accessed: 2016-01-06. 31

[88] C.-W. Hsu, C.-C. Chang, and C.-J. et al. Lin. A practical guide to support vector classification. 2003. 32, 42

[89] P. Refaeilzadeh, L. Tang, and H. Liu. Cross-validation. In *Encyclopedia of Database Systems*, pages 532–538. Springer, 2009. 32

[90] C. Zhang and Y. Ma. *Ensemble machine learning: methods and applications*. Springer, 2012. 33

[91] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2008. 34

[92] L. Breiman, JH Friedman, RA Olshen, and CJ Stone. Classification and regression trees (1984). Monterey, ca: Wadsworth Brooks. 35

[93] D Michie, M Nuñez, V Podgorelec, P Kokol, B Stiglic, and I Rozman. C4. 5: Programs for machine learning, 1993. 35

[94] H. Strasser and C. Weber. On the asymptotic theory of permutation statistics. 1999. 35

[95] Interpreting Ctree output in R. https://stats.stackexchange.com/questions/171301/interpreting-ctree-partykit-output-in-r. Accessed: 2017-10-10. 35

[96] Ensemble methods. http://scikit-learn.org/stable/modules/ensemble.html. Accessed: 2016-12-03. 35, 41

[97] J. Liu, J. Gough, and B. Rost. Distinguishing protein-coding from non-coding RNAs through support vector machines. *PLoS Genetics*, 2(4):e29, Apr 2006. 36, 37

[98] L. Kong, Y. Zhang, Z.-Q. Ye, X.-Q. Liu, S.-Q. Zhao, L. Wei, and G. Gao. CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine. *Nucleic Acids Research*, 35:W345–9, Jul 2007. 36, 37, 38

[99] I. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13):3429–3431, 2003. 38

[100] Human vega68. http://vega.archive.ensembl.org/Homo_sapiens/Info/Index. Accessed:2017-06-18. 40, 41, 44

[101] Human vega68. http://vega.archive.ensembl.org/Mus_musculus/Info/Index. Accessed:2017-06-18. 40, 41, 47

[102] S. F. Altschul, W. Gish, W. Miller, E. W Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410, 1990. 41

[103] M. E Dinger, K. C Pang, T. R Mercer, and J. S Mattick. Differentiating protein-coding and noncoding RNA: challenges and ambiguities. *PLoS computational biology*, 4(11):e1000176, 2008. 42

[104] B. Kummel. Método baseado em aprendizado de máquina para seleção de características para distinção entre rnas não-codificadores longos e rnas codificadores de proteínas. Dissertação de mestrado. Departamento de Ciência da Computação. Universidade de Brasília. 2017. 42

[105] B. D Ripley. The r project in statistical computing. *MSOR Connections. The newsletter of the LTSN Maths, Stats & OR Network*, 1(1):23–25, 2001. 42

[106] Tuning machine learning models using the caret r package. `http://machinelearningmastery.com/tuning-machine-learning-models-using-the-caret-r-package/`. Accessed:2017-06-04. 42

[107] M. B. Clark and J. S. Mattick. Long noncoding rnas in cell biology. In *Seminars in cell & developmental biology*, volume 22, pages 366–376. Elsevier, 2011. 53

[108] M. Cabili, C. Trapnell, L. Goff, M. Koziol, B. Tazon-Vega, A. Regev, and J. Rinn. Integrative annotation of human large intergenic noncoding RNAs reveals global properties and specific subclasses. *Genes & development*, 25(18):1915–1927, 2011. 53

[109] M. Mokhtarzad, F. Eskandari, N. J. Vanjani, and A. Arabasadi. Drought forecasting by ann, anfis, and svm and comparison of the models. *Environmental Earth Sciences*, 76(21):729, 2017. 55

[110] F. Thiebaut, C. A Rojas, C. Grativol, E. P Calixto, M. Motta, H. GF Ballesteros, B. Peixoto, B. NS de Lima, L. M Vieira, M. E. Walter, et al. Roles of non-coding rna in sugarcane-microbe interaction. *Non-Coding RNA*, 3(4):25, 2017. 55