Πανεπιστήμιο Θεσσαλίας
Πολυτεχνική Σχολή
Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Ηλεκτρονικών Υπολογιστών

Υπολογιστική Ανάλυση Γενωμικών Αλληλουχιών

Μεταπτυχιακή Διατριβή

Κωνσταντίνος Γ. Λιάκος

Επιβλέποντες Καθηγητές: Χατζηγεωργίου Άρτεμις
Καθηγήτρια

Χούστη Αικατερίνη
Καθηγήτρια

Ποταμιάνος Γεράσιμος
Αναπληρωτής Καθηγητής

Βόλος, Ιούνιος 2016

UNIVERSITY OF THESSALY
DEPARTEMENT OF ELECTRICAL AND COMPUTER ENGINEERING

Computational Analysis of Genomic Sequences

Master Thesis

Konstantinos G. Liakos

Supervising Professors: Hatzigeorgiou Artemis
Professor

Housti Catherine
Professor

Potamianos Gerasimos
Associate Professor

Approved by the three-member inquiry committee at June 17, 2016

.....	.....	.....
Hatzigeorgiou Artemis Professor	Housti Catherine Professor	Potamianos Gerasimos Associate Professor

To my family & my friends

Ευχαριστίες

Με την ολοκλήρωση της παρούσας εργασίας, θα ήθελα να ευχαριστήσω θερμά τα άτομα που συνέβαλαν με τη συνεισφορά τους, για την βελτιστοποίηση και ολοκλήρωση της μεταπτυχιακής διατριβής μου. Συγκεκριμένα τη καθηγητριά μου, τη Χατζηγεωργίου Άρτεμις και την ερευνητική της ομάδα, που αποτελείται από, τους μεταδιδακτορικούς Βλάχο Ιωάννη, Γεωργακίλα Γεώργιο, τους διδακτορικούς, Παρασκευοπούλου Μαρία, Καραγκούνη Δήμητρα και Ταστσόγλου Σπύρο για τη βοηθειά τους, την εμπιστοσύνη που επέδειξαν στο πρόσωπό, την άρτια συνεργασία μας, τις ουσιώδεις υποδείξεις και επεξηγήσεις, που τελειοποίησαν τη παρούσα διατριβή. Τέλος, θα ήθελα να ευχαριστήσω ολόψυχα την οικογενειά μου, τη κοπέλα μου και τους φίλους μου, που όλα αυτά τα χρόνια δείχνουν την αμέριστη συμπαράστασή τους ως προς το πρόσωπό μου τόσο στο κομμάτι των σπουδών όσο και της ζωής και βρίσκονται πάντα δίπλα μου.

Λιάκος Κωνσταντίνος
Αθήνα, 2016

Contents

List of Tables

List of Figures

Περίληψη

Στην παρούσα μεταπτυχιακή διατριβή, ασχολούμαστε με το ανθρώπινο γονίδιο και με το πού παράγεται η πρωτεΐνη στο ανθρώπινο γονιδίωμα. Συγκεκριμένα, ο στόχος μας είναι να ταυτοποιήσουμε αν όντως οι περιοχές που παρουσιάζονται ως κωδικές περιοχές του γονιδιώματος, οι οποίες είναι οι περιοχές που παράγεται πρωτεΐνη στο ανθρώπινο γονιδίωμα είναι ή δεν είναι κωδικές και αντίστοιχα αν όντως οι περιοχές που χαρακτηρίζονται ως μη κωδικές, που είναι μέρη του ανθρώπινου οργανισμού που δε συντίθεται πρωτεΐνη, είναι ή δεν είναι μη κωδικές αντίστοιχα, δηλαδή παράγουν ή δεν παράγουν πρωτείνη στα κύτταρα του ανθρώπινου οργανισμού. Ο τρόπος με τον οποίο γίνεται η ανίχνευση και η ταυτοποίηση αποτελείται από διάφορα στάδια και διαδικασίες που περιγράφονται στα επόμενα κεφάλαια. Ένα πολύ σημαντικό κομμάτι της συγκεκριμένης εργασίας, είναι ο εντοπισμός και η σύγκριση των τελικών αποτελεσμάτων, ανάμεσα σε κάποια από τα πιο γνωστά προγράμματα που υπάρχουν για την ανίχνευση και ταυτοποίηση των γονιδίων, που ευθύνονται για την παραγωγή πρωτεΐνης στον άνθρωπο και έπειτα ο τρόπος λειτουργίας τους, το τι προσφέρουν στο χρήστη και πώς μπορούν να βελτιωθούν. Το τελικό κομμάτι της διατριβής είναι η παρουσίαση της δικιάς μας πρότασης για την ανίχνευση και την ταυτοποίηση των κωδικών και μη περιοχών, με τη χρήση ενός εργαλείου και νέων σύγχρονων αλγορίθμων που έχουμε δημιουργήσει. Τέλος έχουμε τη σύγκριση των αποτελεσμάτων μας έναντι των πιο γνωστών εργαλείων που υπάρχουν.

Abstract

In this thesis, we deal with the human gene and which regions of the human genome produce protein. Specifically, our goal is to identify if areas that seem like gene coding regions(mRNAs), which are the areas that produce proteins in the human genome, are indeed coding or not, and reversely, if areas identified as non-coding, namely, parts of the human genome that don't produce protein are in fact non-coding (ncRNAs). Detection and identification of coding and non-coding regions respectively consists of several stages and procedures which are described in the following chapters. A very important part of this work was to identify and compare the final results among some of the most known programs (state of the art) that exist for the detection and identification of genes which are responsible for the production of proteins in humans. Also find the way these programs operate and what they offer to the user and suggest improvements. The final part of this thesis presents our own integrated pipeline for the distinguish of coding genes (mRNAs) from non – coding genes (ncRNAs) and more specific from long non - coding genes (lincRNAs). In the end we have the comparison of our results versus most common available TIS tools.

Chapter 1
Introduction to Biology

1.1 The Central Dogma of Molecular Biology

Before proceeding to the main subject this master's thesis deals with, it would be good to remember the central dogma of molecular biology.

The central dogma of molecular biology is an explanation of the flow of genetic information within a biological system, as shown in Figure 1.1. It was first stated by Francis Crick in 1956 and then re-stated in a publication in 1970 (Crick 1970) and is currently the bible of molecular biology.

The central dogma of molecular biology deals with the detailed relationship between DNA, RNA and PROTEIN. RNA is produced from DNA and then PROTEIN is produced from RNA. Even the main functions of DNA, RNA and PROTEIN production are depicted in the central dogma; the DNA replicates with the help of DNA polymerase, it is transcribed to RNA with the help of RNA polymerase and the RNA is translated to PROTEIN by the Ribosome.

During the following decades the molecular basis of the central dogma was elucidated (Watson 2007) and special routes that don't exist in all species were included. These are RNA replication and reverse transcription and we will refer to them briefly in 1.3 About RNA.

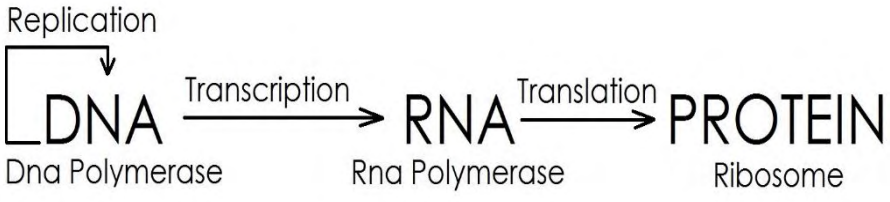


Figure 1.1: The Central Dogma of Molecular Biology.

1.2 About DNA

Deoxyribonucleic acid, else known as DNA, is a molecule that carries most of the genetic instructions used in the development and functioning of all known living organisms and many viruses. DNA is a nucleic acid; alongside proteins and carbohydrates, nucleic acids compose the three major macromolecules essential for all known forms of life. Most DNA molecules consist of two biopolymer strands coiled around each other to form a double helix (Berg, Tymoczko et al. 2007). The two DNA strands are known as polynucleotides since they are composed of simpler units called nucleotides. Each nucleotide is composed of a nitrogen-containing nucleobase—either guanine (G), adenine (A), thymine (T), or cytosine (C)—as well as a monosaccharide sugar called 2-deoxyribose and a phosphate group. The nucleotide sugars are joined together by phosphate groups that form phosphodiester bonds between the third and fifth carbon atoms of adjacent sugar rings, consequently composing the DNA backbone. Attached to each sugar is one of the four types of nucleobases and it is the sequence of them that encodes biological information. According to base pairing rules (A with T, and C with G), hydrogen bonds bind the nucleobases of the two separate polynucleotide strands to make double-stranded DNA.
In a double helix, the direction of the nucleotides in one strand is opposite to their direction in the other strand: the strands are antiparallel. The asymmetric ends of DNA strands are called the 5′ (five prime) and 3′ (three prime) ends, with the 5′ end having a terminal phosphate group and the 3′ end a terminal hydroxyl group.

The DNA double helix is stabilized primarily by two forces: hydrogen bonds between nucleotides and base-stacking interactions among aromatic nucleobases. In the aqueous environment of the cell, the conjugated π bonds of nucleotide bases align perpendicular to the axis of the DNA molecule, minimizing their interaction with the solvation shell and therefore, the Gibbs free energy.

Double-strand DNA contains a major and a minor groove, created by the way each pair of bases is lined up after the other (Berg, Tymoczko et al. 2007). Additionally, it can form various types of helices; the most common and first one to be identified is B-DNA, a right-handed helix with wide and deep grooves that permit easier interaction with DNA-binding proteins. A-DNA is also right-handed but the ribose molecules are more tightly packed, leading to a more compact conformation with a narrow major groove and a shallow minor groove. A-DNA is favored in dehydrated DNA, or in double-strand RNA, or in DNA-RNA hybrid molecules. Lastly, Z-DNA is an even tighter, left-handed,

helix conformation that is observed only in short oligonucleotide chains with a specific sequence; Its biological role is still under question.

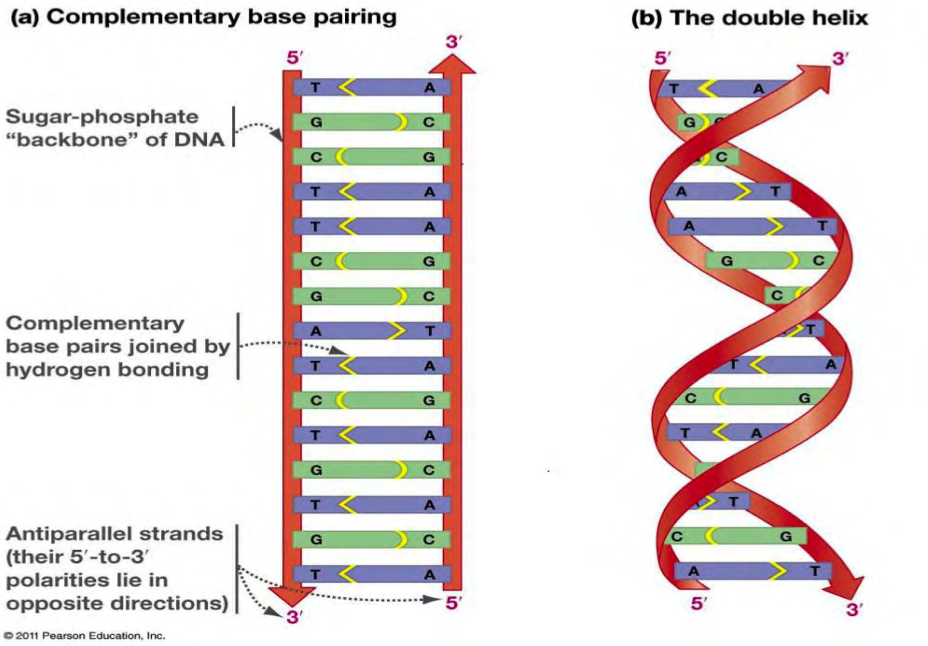


Figure 1.2: The antiparallel DNA strands form a double helix [4].

DNA is well-suited for biological information storage. The DNA backbone is resistant to cleavage, and both strands of the double-stranded structure store the same biological information, "immediately suggesting a possible copying mechanism for the genetic material", as Watson and Crick pointed out in 1953 (Watson and Crick 1953); biological information is replicated as the two strands are separated. A significant portion of DNA (more than 98% for humans) is non-coding, meaning that these sections do not serve as patterns for protein sequences.

Within cells, DNA is organized into long structures called chromosomes. During cell division these chromosomes are duplicated in the process of DNA replication, providing each cell its own complete set of chromosomes. Eukaryotic organisms (Animalia, Plantae, Fungi, and Protista) store most of their DNA inside the cell nucleus and some of their DNA in organelles, such as mitochondria or chloroplasts (Russell 2010). In contrast, prokaryotes (Bacteria and Archaea) store their DNA only in a roughly defined space in the cytoplasm called nucleoid. Within the chromosomes, chromatin proteins such as histones compact and organize DNA. These compact structures guide the interactions between DNA and other proteins, helping control which parts of the DNA are transcribed.

Historically, DNA was first identified and isolated by Friedrich Miescher in 1871, and the double helix structure of DNA was first discovered by James Watson and Francis Crick, using experimental data

collected by Rosalind Franklin and Maurice Wilkins. Its two helical chains are coiled around the same axis, each with a pitch of 34 ångströms and a radius of 10 ångströms. Although each individual repeating unit is very small, DNA polymers can be very large molecules containing millions of nucleotides. For instance, the largest human chromosome, chromosome number 1, consists of approximately 220 million base pairs and is 85 mm long.

Apart from its uses in molecular biology and genetic engineering, DNA is also used by researchers as a molecular tool to explore physical laws and theories, such as the ergodic theorem and the theory of elasticity. The unique material properties of DNA have made it an attractive molecule for material scientists and engineers interested in micro- and nano-fabrication. Among notable advances in this field are DNA origami and DNA-based hybrid materials.

The obsolete synonym "desoxyribonucleic acid" may occasionally be encountered, for example, in pre-1953 genetics.

1.3 About RNA

Ribonucleic acid, known as RNA is a polymeric molecule. It is implicated in various biological roles in coding, regulation, and expression of genes. Like DNA, RNA is assembled as a chain of nucleotides, but unlike DNA it is more often found in nature as a single-strand folded unto itself, rather than a paired double-strand. Organisms use messenger RNA known as mRNA to convey genetic information that directs synthesis of specific proteins. Many viruses encode their genetic information using an RNA genome.

Some RNA molecules play an active role within cells by catalyzing biological reactions, controlling gene expression, or sensing and communicating responses to cellular signals. One of these active processes is protein synthesis, a universal function whereby mRNA molecules direct the assembly of proteins on ribosomes. This process uses transfer RNA known as tRNA molecules to deliver amino acids to the ribosome, where ribosomal RNA, known as rRNA links amino acids together to form proteins.

Each nucleotide in RNA contains a ribose sugar, with carbons numbered 1' through 5'. A base is attached to the 1' position, in general, adenine (A), cytosine (C), guanine (G), or uracil (U). Adenine and guanine are purines, cytosine and uracil are pyrimidines. A phosphate group is attached to the 3' position of one ribose and the 5' position of the next. The phosphate groups have a negative charge each at physiological pH, making RNA a charged molecule. The bases form hydrogen bonds between C and G, between A and U and between G and U (Lee and Gutell 2004). However, other interactions are possible,

such as a group of adenine bases binding to each other in a bulge (Barciszewski, Clark et al. 1999), or the GNRA tetraloop that has a guanine–adenine base-pair (Lee and Gutell 2004).

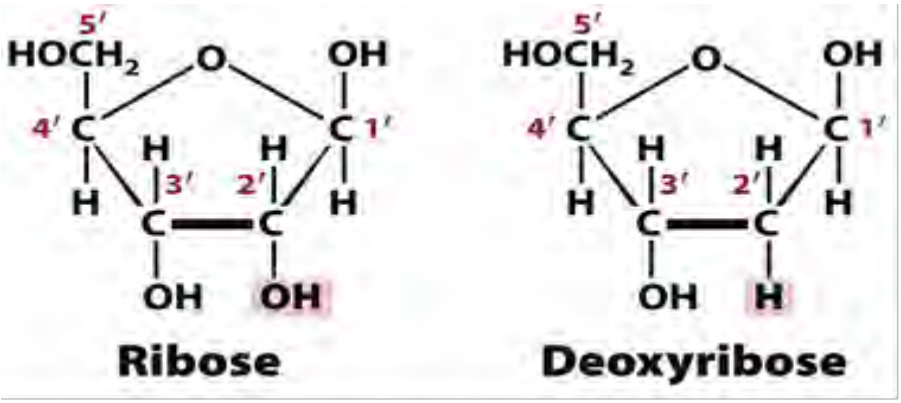


Figure1.3: Ribose & Deoxyribose [3].

As shows in Figure 1.3, an important structural feature of RNA that distinguishes it from DNA is the presence of a hydroxyl group at the 2' position of the ribose sugar. The presence of this functional group causes the helix to adopt the A-form geometry rather than the B-form most commonly observed in DNA (Sedova and Banavali 2015). This results in a very deep and narrow major groove and a shallow and wide minor groove (Hermann and Patel 2000). A second consequence of the presence of the 2'-hydroxyl group is that in conformation ally flexible regions of an RNA molecule, it can chemically attack the adjacent phosphodiester bond to cleave the backbone (Mikkola, Stenman et al. 1999).

Although RNA is transcribed with only four bases, these bases and attached sugars can be modified in numerous ways as the RNAs mature. Pseudouridine, in which the linkage between uracil and ribose is changed from a C–N bond to a C–C bond, and ribothymidine are found in various places(Yu and Morrow 2001). Another notable modified base is hypoxanthine, a deaminated adenine base whose nucleoside is called inosine. Inosine plays a key role in the wobble hypothesis of the genetic code, that is, the matching of RNA base pairs not according to the classic base pair rules(Elliott and Trewyn 1984).

There are more than 100 other naturally occurring modified nucleosides(Cantara, Crain et al. 2011). The greatest structural diversity of modifications can be found in tRNA, while pseudouridine and nucleosides with 2'-O-methylribose often present in rRNA are the most common (Kiss 2001). The specific roles of many of these modifications in RNA are not fully understood. However, it is notable that, in ribosomal

RNA, many of the post-transcriptional modifications occur in highly functional regions, such as the peptidyl transferase center and the subunit interface, implying that they are important for normal function (King, Liu et al. 2003).

The functional form of single-stranded RNA molecules, just like proteins, frequently requires a specific tertiary structure. The scaffold for this structure is provided by secondary structural elements that are hydrogen bonds within the molecule. This leads to several recognizable domains of secondary structure like hairpin loops, bulges, and internal loops (Mathews, Disney et al. 2004). Since RNA is negatively charged, metal ions such as Mg^{2+} are needed to stabilize many secondary and tertiary structures.

Synthesis of RNA is usually catalyzed by an enzyme—RNA polymerase—using DNA as a template, a process known as transcription. Initiation of transcription begins with the binding of the enzyme to a promoter sequence in the DNA, usually found "upstream" of a gene. The DNA double helix is unwound by the helicase activity of the enzyme. The enzyme then progresses along the template strand in the 3' to 5' direction, synthesizing a complementary RNA molecule with elongation occurring in the 5' to 3' direction. The DNA sequence also dictates where termination of RNA synthesis will occur (Nudler and Gottesman 2002).

Primary transcript RNAs are often modified by enzymes after transcription. For example, a poly (A) tail and a 5' cap are added to eukaryotic pre-mRNA and introns are removed by the spliceosome.

There are also a number of RNA-dependent RNA polymerases that use RNA as their template for synthesis of a new strand of RNA. For instance, a number of RNA viruses use this type of enzyme to replicate their genetic material (Hansen, Long et al. 1997). Also, RNA-dependent RNA polymerase is part of the RNA interference pathway in many organisms(Ahlquist 2002).

Coding and non-coding RNA species

Messenger RNA or else mRNA, is the RNA class that carries information from DNA to the ribosome, the cellular site of protein synthesis or translation. The coding sequence of the mRNA determines the amino acid sequence(Cooper and Hausman 2004). However, many RNAs do not code for protein, about 97% of the transcriptional output is non-protein-coding in eukaryotes (Mattick 2001, Mattick 2003).

These so-called **non-coding RNAs**, or ncRNA, can be encoded by their own genes, but can also derive from mRNA introns (St Laurent, Shtokalo et al. 2012). The most prominent examples of non-coding RNAs are transfer RNA or tRNA and ribosomal RNA or rRNA, both of which are

involved in the process of translation. There are also non-coding RNAs involved in gene regulation, RNA processing and other roles. Certain RNAs are able to catalyze chemical reactions such as cutting and ligating other RNA molecules (Rossi 2004), and the catalysis of peptide bond formation in the ribosome. These are known as ribozymes (Nissen, Hansen et al. 2000).

In translation, the messenger RNA carries information about a protein sequence to the ribosomes, the protein synthesis factories in the cell. It is coded so that every three nucleotides or else a codon correspond to one amino acid. In eukaryotic cells, once precursor mRNA or pre-mRNA has been transcribed from DNA, it is processed to mature mRNA. This removes its introns—non-coding sections of the pre-mRNA. The mRNA is then exported from the nucleus to the cytoplasm, where it is bound to ribosomes and translated into its corresponding protein form with the help of tRNA. In prokaryotic cells, which do not have nucleus and cytoplasm compartments, mRNA can bind to ribosomes while it is being transcribed from DNA. After a certain amount of time the message degrades into its component nucleotides with the assistance of ribonucleases (Cooper and Hausman 2004).

Transfer RNA is a small RNA chain of about 80 nucleotides that transfers a specific amino acid to a growing polypeptide chain at the ribosomal site of protein synthesis during translation. It has sites for amino acid attachment and an anticodon region for codon recognition that binds to a specific sequence on the messenger RNA chain through hydrogen bonding. [26]

Ribosomal RNA is the catalytic component of the ribosomes. Eukaryotic ribosomes contain four different rRNA molecules: 18S, 5.8S, 28S and 5S rRNA. S, Svedberg, is a non-SI unit of sedimentation, practically the time a molecule needs to settle at the bottom of a test tube under an acceleration of 10^7 m/s^2, in a centrifuge. Svedberg units are not directly additive, as they depend on the particle mass, shape and density (Correia and Stafford 2015). Three of the rRNA molecules are synthesized in the nucleolus, and one is synthesized elsewhere. In the cytoplasm, ribosomal RNA and protein combine to form a nucleoprotein complex called a ribosome. The ribosome binds mRNA and carries out protein synthesis. Several ribosomes may be attached to a single mRNA at any time (Cooper and Hausman 2004). Most RNA found in a typical eukaryotic cell is rRNA.

Transfer-messenger RNA or else tmRNA is found in many bacteria and plastids. It tags proteins encoded by mRNAs that lack stop codons for degradation and prevents the ribosome from stalling (Gueneau de Novoa and Williams 2004).

Regulatory RNAs:

Several types of RNA can down-regulate gene expression by being complementary to a part of an mRNA or a gene's DNA. **microRNAs** are found in eukaryotes and act through RNA interference or RNAi, where an effector complex of miRNA and enzymes can cleave complementary mRNA, block the mRNA from being translated, or accelerate its degradation (Guo, Ingolia et al. 2010).

Small interfering RNAs or siRNA are often produced by breakdown of viral RNA, but there can also be endogenous sources of siRNAs (Vazquez, Vaucheret et al. 2004). siRNAs act through RNA interference in a fashion similar to miRNAs. Some miRNAs and siRNAs can cause genes they target to be methylated, thereby decreasing or increasing transcription of those genes (Sontheimer and Carthew 2005, Pushparaj, Aarthi et al. 2008).

Animals have **Piwi-interacting RNAs** or piRNA that are active in germline cells and are thought to be a defense against transposons and play a role in gametogenesis (Girard, Sachidanandam et al. 2006, Horwich, Li et al. 2007).

Many prokaryotes have **CRISPR RNAs**, a regulatory system similar to RNA interference (Horvath and Barrangou 2010). Antisense RNAs are widespread; most downregulate a gene, but a few are activators of transcription (Wagner, Altuvia et al. 2002). One way antisense RNA can act is by binding to an mRNA, forming double-stranded RNA that is enzymatically degraded. There are many long non-coding RNAs that regulate genes in eukaryotes (Amaral and Mattick 2008), one such RNA is Xist, which coats one X chromosome in female mammals and inactivates it (Heard, Mongelard et al. 1999).

An mRNA may contain regulatory elements itself, such as riboswitches, in the 5' untranslated region or 3' untranslated region, these cis-regulatory elements regulate the activity of that mRNA. The untranslated regions can also contain elements that regulate other genes (Scotto and Assoian 1993, Batey 2006).

RNA processing species:

Many RNAs are involved in modifying other RNAs. Introns are spliced out of pre-mRNA by spliceosomes, which contain several small nuclear RNAs known as snRNA(Berg, Tymoczko et al. 2007) or the introns can be ribozymes that are spliced by themselves (Steitz and Steitz 1993). RNA can also be altered by having its nucleotides modified to other nucleotides than A, C, G and U. In eukaryotes, modifications of RNA nucleotides are in general directed by small nucleolar RNAs, found in the nucleolus and cajal bodies. snoRNAs associate with enzymes and guide them to a spot on an RNA by basepairing to that RNA. These enzymes

then perform the nucleotide modification. rRNAs and tRNAs are extensively modified, but snRNAs and mRNAs can also be the target of base modification (Xie, Zhang et al. 2007). RNA can also be methylated (Cavaille, Nicoloso et al. 1996).

RNA genomes:

Much like DNA, RNA can carry genetic information. RNA viruses have genomes composed of RNA that encodes a number of proteins. The viral genome is replicated by some of those proteins, while other proteins protect the genome as the virus particle moves to a new host cell. Viroids are another group of pathogens, but they consist only of RNA, do not encode any protein and are replicated by a host plant cell's polymerase (Daros, Elena et al. 2006).

In reverse transcription, reverse transcribing viruses replicate their genomes by reverse transcribing DNA copies from their RNA. Either the same or the opposite (antisense) DNA chain of these copies can then be transcribed to new RNAs (Madigan, Madigan et al. 2009). Retrotransposons also spread by copying DNA and RNA from one another (Kalendar, Vicient et al. 2004), and telomerase contains an RNA that is used as template for building the ends of eukaryotic chromosomes (Podlevsky, Bley et al. 2008).

Double-stranded RNA or else dsRNA is RNA with two complementary strands, similar to the DNA found in all cells. dsRNA forms the genetic material of some viruses like double-stranded RNA viruses. Double-stranded RNA such as viral RNA or siRNA can trigger RNA interference in eukaryotes, as well as interferon response in vertebrates(Blevins, Rajeswaran et al. 2006, Whitehead, Dahlman et al. 2011).

It is worth emphasizing again the fact that RNA can carry biological information as well as have an enzymatic activity; it is a less stable macromolecule than DNA and it can catalyze less diverse reactions that proteins. Therefore, although today it is positioned as an intermediate of information flow between DNA and proteins, the "RNA world hypothesis" is being shaped since the 1950s, suggesting that self-replicating RNA molecules are precursors of all current life forms on Earth (Berg, Tymoczko et al. 2007, Robertson and Joyce 2012).

1.4 About Protein

Proteins are macromolecules, consisting of one or more long chains of amino acid residues. Proteins perform a vast array of functions within living organisms, including catalyzing metabolic reactions,

replicating DNA, responding to stimuli, and transporting molecules from one location to another. Proteins differ from one another primarily in their sequence of amino acids, which is dictated by the nucleotide sequence of their genes, and which usually results in folding of the protein into a specific three-dimensional structure that determines its activity.

A linear chain of amino acid residues is called a polypeptide(Berg, Tymoczko et al. 2007). A protein contains at least one long polypeptide. Short polypeptides, containing less than about 20-30 residues, are rarely considered to be proteins and are commonly called peptides, or sometimes oligopeptides. The individual amino acid residues are bonded together by peptide bonds and adjacent amino acid residues. The sequence of amino acid residues in a protein is defined by the sequence of a gene, which is encoded in the genetic code. In general, the genetic code specifies 20 standard amino acids, however, in certain organisms the genetic code can include selenocysteine and—in certain archaea—pyrrolysine (Rother and Krzycki 2010). Shortly after or even during synthesis, the residues in a protein are often chemically modified by posttranslational modifications (PTMs), which alter the physical and chemical properties, folding, stability, activity, and ultimately, the function of the proteins. Sometimes proteins have non-peptide groups attached, which can be called prosthetic groups or cofactors(Khoury, Baliban et al. 2011). Proteins can also work together to achieve a particular function, and they often associate to form stable protein complexes.

Once formed, proteins only exist for a certain period of time and are then degraded and recycled by the cell's machinery through the process of protein turnover (Toyama and Hetzer 2013). A protein's lifespan is measured in terms of its half-life and covers a wide range. They can exist for minutes or years with an average lifespan of 1–2 days in mammalian cells. Abnormal and/or misfolded proteins are degraded more rapidly either due to being targeted for destruction or due to being unstable.

Like other biological macromolecules such as polysaccharides and nucleic acids, proteins are essential parts of organisms and participate in virtually every process within cells. Many proteins are enzymes that catalyze biochemical reactions and are vital to metabolism. Proteins also have structural or mechanical functions, such as actin and myosin in muscle and the proteins in the cytoskeleton, which form a system of scaffolding that maintains cell shape. Other proteins are important in cell signaling, immune responses, cell adhesion, and the cell cycle. Proteins are also necessary in animals' diets, since animals cannot synthesize all the amino acids they need and must obtain essential amino acids from food. Through the process of

digestion, animals break down ingested protein into free amino acids that are then used in metabolism.

Proteins may be purified from other cellular components using a variety of techniques such as ultracentrifugation, precipitation, electrophoresis, and chromatography; the advent of genetic engineering has made possible a number of methods (Labrou 2014) to facilitate purification. Methods commonly used to study protein structure and function include immunohistochemistry, site-directed mutagenesis, X-ray crystallography, nuclear magnetic resonance and mass spectrometry.

1.5 DNA Replication

DNA replication is the process of producing two identical replicas from one original DNA molecule. This biological process occurs in all living organisms and is the basis for biological inheritance. DNA is made up of two strands and each strand of the original DNA molecule serves as a template for the production of the complementary strand, a process referred to as semiconservative replication. Cellular proofreading and error-checking mechanisms ensure near perfect fidelity for DNA replication (Berg, Tymoczko et al. 2007).

In a cell, DNA replication begins at specific locations, or origins of replication, in the genome. Unwinding of DNA at the origin and synthesis of new strands results in replication forks growing bidirectional from the origin. A number of proteins are associated with the replication fork which helps in terms of the initiation and continuation of DNA synthesis. Most prominently, DNA polymerase synthesizes the new DNA by adding complementary nucleotides to the template strand.

DNA replication can also be performed in vitro (artificially, outside a cell). DNA polymerases isolated from cells and artificial DNA primers can be used to initiate DNA synthesis at known sequences in a template DNA molecule. The polymerase chain reaction (PCR), a common laboratory technique, cyclically applies such artificial synthesis to amplify a specific target DNA fragment from a pool of DNA.

The pairing of complementary bases in DNA through hydrogen bonding means that the information contained within each strand is redundant. The nucleotides on a single strand can be used to reconstruct nucleotides on a newly synthesized partner strand(Alberts 2008).

DNA polymerases are a family of enzymes that carry out all forms of DNA replication (Berg, Tymoczko et al. 2007). DNA polymerases in general cannot initiate synthesis of new strands, but can only extend an existing DNA or RNA strand paired with a template strand. To begin synthesis, a short fragment of RNA, called a primer, must be created and paired with the template DNA strand.

DNA polymerase adds a new strand of DNA by extending the 3' end of an existing nucleotide chain, adding new nucleotides matched to the template strand one at a time via the creation of phosphodiester bonds. The energy for this process of DNA polymerization comes from hydrolysis of the high-energy phosphate or known as phosphoanhydride bonds between the three phosphates attached to each unincorporated base. Free bases with three attached phosphate groups are called nucleoside triphosphates. When a nucleotide is being added to a growing DNA strand, the formation of a phosphodiester bond between the proximal phosphate of the nucleotide to the growing chain is accompanied by hydrolysis of a high-energy phosphate bond with release of the two distal phosphates as a pyrophosphate. Enzymatic hydrolysis of the resulting pyrophosphate into inorganic phosphate consumes a second high-energy phosphate bond and renders the reaction effectively irreversible.

As described in 1.2, DNA strands have a directionality, and the different ends of a single strand are called the 3' end and the 5' end. By convention, if the base sequence of a single strand of DNA is given, the left end of the sequence is 5' end, while the right end of the sequence is the 3' end. The strands of the double helix are anti-parallel with one being 5' to 3', and the opposite strand 3' to 5'. These terms refer to the carbon atom in deoxyribose to which the next phosphate in the chain attaches. Directionality has consequences in DNA synthesis, because DNA polymerase can synthesize DNA in only one direction by adding nucleotides to the 3' end of a DNA strand. Therefore, replication goes on normally on the 3' to 5' strand, resulting in the making of a new strand called leading strand. On the other hand, many more RNA primers bind on the 5' to 3' strand and elongation happens uncontinuously with 5' to 3' directionality. The occurring DNA fragments, called Okazaki fragments, are then stiched together by DNA ligase and finally form the so-called lagging strand, as seen on the figure (Berg, Tymoczko et al. 2007).

In general, DNA polymerases are highly accurate, with an intrinsic error rate of less than one mistake for every 10^7 nucleotides added (McCulloch and Kunkel 2008). In addition, some DNA polymerases also have proofreading ability; they can remove nucleotides from the end of a growing strand in order to correct mismatched bases. Finally, post-replication mismatch repair mechanisms monitor the DNA for errors,

being capable of distinguishing mismatches in the newly synthesized DNA strand from the original strand sequence. Together, these three discrimination steps enable replication fidelity of less than one mistake for every 10^9 nucleotides added.

In the sense that DNA replication must occur if genetic material is to be provided for the progeny of any cell, whether somatic or reproductive, the copying from DNA to DNA arguably is the fundamental step in the central dogma. A complex group of proteins called the replisome performs the replication of the information from the parent strand to the complementary daughter strand (Figure 1.4). The replisome comprises a helicase:

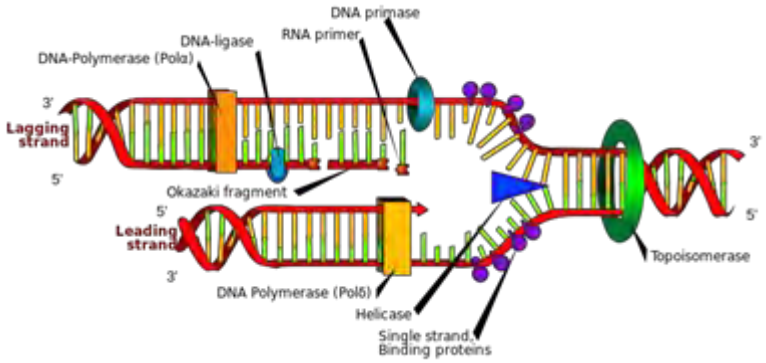


Figure1.4: Unwinding of the superhelix (Ruiz Mariana, 01-24-2007, Public Domain).

that unwinds the superhelix as well as the double-stranded DNA helix to create a replication fork, SSB protein that binds open the double-stranded DNA to prevent it from reassociating, RNA primase that adds a complementary RNA primer to each template strand as a starting point for replication, DNA polymerase III that reads the existing template chain from its 3' end to its 5' end and adds new complementary nucleotides from the 5' end to the 3' end of the daughter chain, DNA polymerase I that removes the RNA primers and replaces them with DNA, DNA ligase that joins the two Okazaki fragments with phosphodiester bonds to produce a continuous chain. This process typically takes place during S phase of the cell cycle.

1.6 RNA Transcription

Transcription is the process by which the information contained in a section of DNA is replicated in the form of a newly assembled piece of messenger RNA known as mRNA (Berg, Tymoczko et al. 2007). Enzymes facilitating the process include RNA polymerase and transcription factors. In eukaryotic cells the primary transcript is the pre-mRNA. Pre-mRNA must be processed for translation to proceed. Processing includes

the addition of a 5' cap and a poly-A tail to the pre-mRNA chain, followed by splicing.

5' cap is a methylated guanine making a rare 5'-5' triphosphate bond on the 5' end. This cap provides stability to the mRNA by guarding its 5' end from nuclease or phospatase activity and additionally it enhances its translation. These are also believed to be the roles of the long adenine chain (polyA-tail) that is added at the 3' end.

Splicing is the process during which the spliceosome, a complex molecular machine at the nucleus removes introns from a pre-mRNA and combines introns, forming the processed mRNA that can be translated into protein. Alternative splicing occurs when appropriate, utilizing different exons to increase the diversity of the proteins that any single mRNA can produce. The product of the entire transcription process that began with the production of the pre-mRNA chain, is a mature mRNA chain.

1.7 Protein Translation

In molecular biology and genetics, translation is the process in which cellular ribosomes create proteins. In translation, mRNA is decoded by a ribosome to produce a specific amino acid chain, or polypeptide. The polypeptide later folds into an active protein and performs its functions in the cell. The ribosome facilitates decoding by inducing the binding of complementary tRNA anticodon sequences to mRNA codons. The tRNAs carry specific amino acids that are chained together into a polypeptide as the mRNA passes through and is "read" by the ribosome. The entire process is a part of gene expression. In brief, translation proceeds in four phases and is shown in Figure 1.5:

- Initiation:
  - The ribosome assembles around the target mRNA. The first tRNA is attached at the start codon.
- Elongation:
  - The tRNA transfers an amino acid to the tRNA corresponding to the next codon.
- Translocation:
  - The ribosome then moves (translocates) to the next mRNA codon to continue the process, creating an amino acid chain.
- Termination:
  - When a stop codon is reached, the ribosome releases the polypeptide.

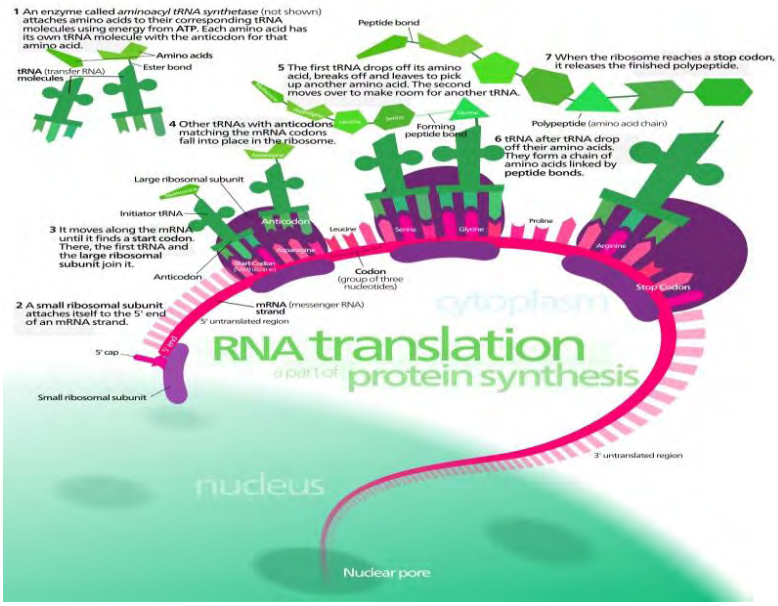


Figure 1.5: RNA translation protein synthesis (Kalvinsong, 12-14-2012, Creative Commons)

In bacteria, translation occurs in the cell's cytoplasm, where the large and small subunits of the ribosome bind to the mRNA. In eukaryotes, translation occurs in the cytosol or across the membrane of the endoplasmic reticulum. In many instances, the entire ribosome/mRNA complex binds to the outer membrane of the rough endoplasmic reticulum (ER); the newly created polypeptide is stored inside the ER for later vesicle transport and secretion outside of the cell.

The basic process of protein production is addition of one amino acid at a time to the end of a protein. This operation is performed by a ribosome. The choice of amino acid type to add is determined by an mRNA molecule. Each amino acid added is matched to a three nucleotide subsequence of the mRNA. For each such triplet possible, the corresponding amino acid is accepted. The successive amino acids added to the chain are matched to successive nucleotide triplets in the mRNA. In this way the sequence of nucleotides in the template mRNA chain determines the sequence of amino acids in the generated amino acid chain (Campbell and Reece 2005). Addition of an amino acid occurs at the C-terminus of the peptide and thus translation is said to be amino-to-carboxyl directed.

The mRNA carries genetic information encoded as a ribonucleotide sequence from the chromosomes to the ribosomes (Berg, Tymoczko et al. 2007). The ribonucleotides are "read" by translational machinery in a sequence of nucleotide triplets called codons. Each of those triplets codes for a specific amino acid.

The ribosome molecules translate this code to a specific sequence of amino acids. The ribosome is a multisubunit structure containing rRNA

and proteins. It is the "factory" where amino acids are assembled into proteins. tRNAs are small noncoding RNA chains (74-93 nucleotides) that transport amino acids to the ribosome. tRNAs have a site for amino acid attachment, and a site called an anticodon. The anticodon is an RNA triplet complementary to the mRNA triplet that codes for their cargo amino acid.

Aminoacyl tRNA synthetases (enzymes) catalyze the bonding between specific tRNAs and the amino acids that their anticodon sequences call for. The product of this reaction is an aminoacyl-tRNA. This aminoacyl-tRNA is carried to the ribosome by EF-Tu, where mRNA codons are matched through complementary base pairing to specific tRNA anticodons. Aminoacyl-tRNA synthetases that mispair tRNAs with the wrong amino acids can produce mischarged aminoacyl-tRNAs, which can result in inappropriate amino acids at the respective position in protein. This "mistranslation" of the genetic code naturally occurs at low levels in most organisms, but certain cellular environments cause an increase in permissive mRNA decoding, sometimes to the benefit of the cell (Moghal, Mohler et al. 2014).

The ribosome has three sites for tRNA to bind. They are the aminoacyl site (abbreviated A), the peptidyl site (abbreviated P) and the exit site (abbreviated E). With respect to the mRNA, the three sites are oriented 5' to 3' E-P-A, because ribosomes move toward the 3' end of mRNA. The A site binds the incoming tRNA with the complementary codon on the mRNA. The P site holds the tRNA with the growing polypeptide chain. The E site holds the tRNA without its amino acid. When an aminoacyl-tRNA initially binds to its corresponding codon on the mRNA, it is in the A site. Then, a peptide bond forms between the amino acid of the tRNA in the A site and the amino acid of the charged tRNA in the P site. The growing polypeptide chain is transferred to the tRNA in the A site. Translocation occurs, moving the tRNA in the P site, now without an amino acid, to the E site; the tRNA that was in the A site, now charged with the polypeptide chain, is moved to the P site. The tRNA in the E site leaves and another aminoacyl-tRNA enters the A site to repeat the process (Griffiths 2008).

After the new amino acid is added to the chain, and after the mRNA is released out of the nucleus and into the ribosome's core, the energy provided by the hydrolysis of a GTP bound to the translocase EF-G (in prokaryotes) and eEF-2 (in eukaryotes) moves the ribosome down one codon towards the 3' end. The energy required for translation of proteins is significant. For a protein containing n amino acids, the number of high-energy phosphate bonds required to translate it is 4n-1[citation needed]. The rate of translation varies; it is significantly higher in prokaryotic cells (up to 17-21 amino acid residues per second) than in

eukaryotic cells (up to 6-9 amino acid residues per second) (Ross and Orlowski 1982).

In activation, the correct amino acid is covalently bonded to the correct transfer RNA (tRNA). The amino acid is joined by its carboxyl group to the 3' OH of the tRNA by an ester bond. When the tRNA has an amino acid linked to it, it is termed "charged". Initiation involves the small subunit of the ribosome binding to the 5' end of mRNA with the help of initiation factors (IF). Termination of the polypeptide happens when the A site of the ribosome faces a stop codon (UAA, UAG, or UGA). No tRNA can recognize or bind to this codon. Instead, the stop codon induces the binding of a release factor protein that prompts the disassembly of the entire ribosome/mRNA complex.

Whereas other aspects such as the 3D structure, called tertiary structure, of protein can only be predicted using sophisticated algorithms, the amino acid sequence, called primary structure, can be determined solely from the nucleic acid sequence with the aid of a translation table.

This approach may not give the correct amino acid composition of the protein, in particular if unconventional amino acids such as selenocysteine are incorporated into the protein, which is coded for by a conventional stop codon in combination with a downstream hairpin (SElenoCysteine Insertion Sequence, or SECIS).

There are many computer programs capable of translating a DNA/RNA sequence into a protein sequence. Normally this is performed using the Standard Genetic Code, many bioinformaticians have written at least one such program at some point in their education. However, few programs can handle all the "special" cases, such as the use of the alternative initiation codons. For instance, the rare alternative start codon CTG codes for Methionine when used as a start codon, and for Leucine in all other positions. Figure 1.6 depicts the standard genetic code.

nonpolar polar basic acidic (stop codon)

Standard genetic code

1st base	2nd base U		2nd base C		2nd base A		2nd base G		3rd base
U	UUU	(Phe/F) Phenylalanine	UCU	(Ser/S) Serine	UAU	(Tyr/Y) Tyrosine	UGU	(Cys/C) Cysteine	U
	UUC		UCC		UAC		UGC		C
	UUA	(Leu/L) Leucine	UCA		UAA	Stop (*Ochre*)	UGA	Stop (*Opal*)	A
	UUG		UCG		UAG	Stop (*Amber*)	UGG	(Trp/W) Tryptophan	G
C	CUU		CCU	(Pro/P) Proline	CAU	(His/H) Histidine	CGU	(Arg/R) Arginine	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	(Gln/Q) Glutamine	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	(Ile/I) Isoleucine	ACU	(Thr/T) Threonine	AAU	(Asn/N) Asparagine	AGU	(Ser/S) Serine	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	(Lys/K) Lysine	AGA	(Arg/R) Arginine	A
	AUG[A]	(Met/M) Methionine	ACG		AAG		AGG		G
G	GUU	(Val/V) Valine	GCU	(Ala/A) Alanine	GAU	(Asp/D) Aspartic acid	GGU	(Gly/G) Glycine	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	(Glu/E) Glutamic acid	GGA		A
	GUG		GCG		GAG		GGG		G

Figure 1.6: Standard genetic code

1.8 Reading frame

In molecular biology, a reading frame is a way of dividing the sequence of nucleotides in a nucleic acid (DNA or RNA) molecule into a set of consecutive, non-overlapping triplets. Where these triplets equate to amino acids or stop signals during translation, they are called codons (an example of a reading frame shown in Figure 1.7).

A single strand of a nucleic acid molecule has a phosphoryl end, called the 5′-end, and a hydroxyl or 3′-end. These define the 5'→3' direction. There are three reading frames that can be read in this 5'→3' direction, each beginning from a different nucleotide in a triplet. In a double stranded nucleic acid, an additional three reading frames may be read from the other, complementary strand in the 5'→3' direction along this strand. As the two strands of a double stranded nucleic acid molecule are antiparallel, the 5'→3' direction on the second strand corresponds to the 3'→5' direction along the first strand (Badger and Olsen 1999).

In general, at most one reading frame in a given section of a nucleic acid is biologically relevant (open reading frame, ORF). Some viral transcripts can be translated using multiple, overlapping reading frames (Kawano, Neeley et al. 2013). There is one example of overlapping reading frames in mammalian mitochondrial DNA: coding portions of genes for 2 subunits of ATPase overlap.

DNA encodes protein sequence by a series of three-nucleotide codons. Any given sequence of DNA can therefore be read in six different ways: Three reading frames in one direction (starting at different nucleotides) and three in the opposite direction. However, during transcription, the template DNA strand is transcribed to mRNA in the 3'→5' direction. The resulting 5' to 3' mRNA is single-stranded and therefore only contains three possible reading frames, of which only one is translated. The codons of the mRNA reading frame are translated in the 5'→3' direction into amino acids by a ribosome to produce a polypeptide chain.

An open reading frame (ORF) is a reading frame that has the potential to be transcribed into RNA and translated into protein. It requires a continuous sequence of DNA from a start codon, through a subsequent region which usually has a length that is a multiple of 3 nucleotides, to a stop codon in the same reading frame.

The usage of multiple reading frames leads to the possibility of overlapping genes; there may be many of these in virus, prokaryote, and mitochondrial genomes (Johnson and Chisholm 2004). Some viruses, e.g. Hepatitis B virus and BYDV, use several overlapping genes in different reading frames.

In rare cases, a ribosome may shift from one frame to another during translation of an mRNA (translational frameshift). This causes the first part of the mRNA to be translated in one reading frame, and the latter part to be translated in a different reading frame. This is distinct from a frameshift mutation, as the nucleotide sequence (DNA or RNA) is not altered—only the frame in which it is read.

Figure 1.7: One strand has three possible reading frames. This figure has been designed for the purpose of this thesis.

1.9 Open reading frame

In molecular genetics, an open reading frame (ORF) is the part of a reading frame that has the potential to code for a protein or peptide. An ORF is a continuous stretch of DNA beginning with a start codon, usually methionine (ATG), and ending with a stop codon (the nucleotide

triplets TAA, TAG or TGA in most genomes), example is shows in Figure 1.8. The transcription termination pause site is located after the ORF, beyond the translation stop codon, because if transcription were to cease before the stop codon, an incomplete protein would be made during translation. Normally, inserts which interrupt the reading frame of a subsequent region after the start codon cause frameshift mutation of the sequence and dislocate the sequences for stop codons.

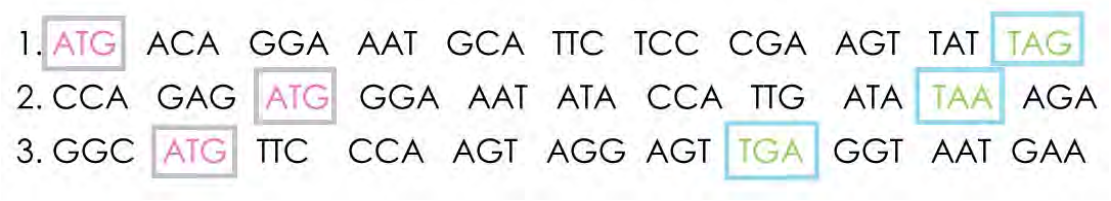


Figure 1.8: Open reading frame ATG – Stop codon TAG, TAA, TGA. This figure has been designed for the purpose of this thesis.

One common use of open reading frames is as one piece of evidence to assist in gene prediction. Long ORFs are often used, along with other evidence, to initially identify candidate protein coding regions in a DNA sequence (Deonier, Tavaré et al. 2005). The presence of an ORF does not necessarily mean that the region is ever translated. For example, in a randomly generated DNA sequence with an equal percentage of each nucleotide, a stop-codon would be expected once every 21 codons. A simple gene prediction algorithm for prokaryotes might look for a start codon followed by an open reading frame that is long enough to encode a typical protein, where the codon usage of that region matches the frequency characteristic for the given organism's coding regions. By itself even a long open reading frame is not conclusive evidence for the presence of a gene.

Since DNA has two anti-parallel strands, an additional three reading frames arise, giving a possible six frame translations as is shown in Figure 1.9.

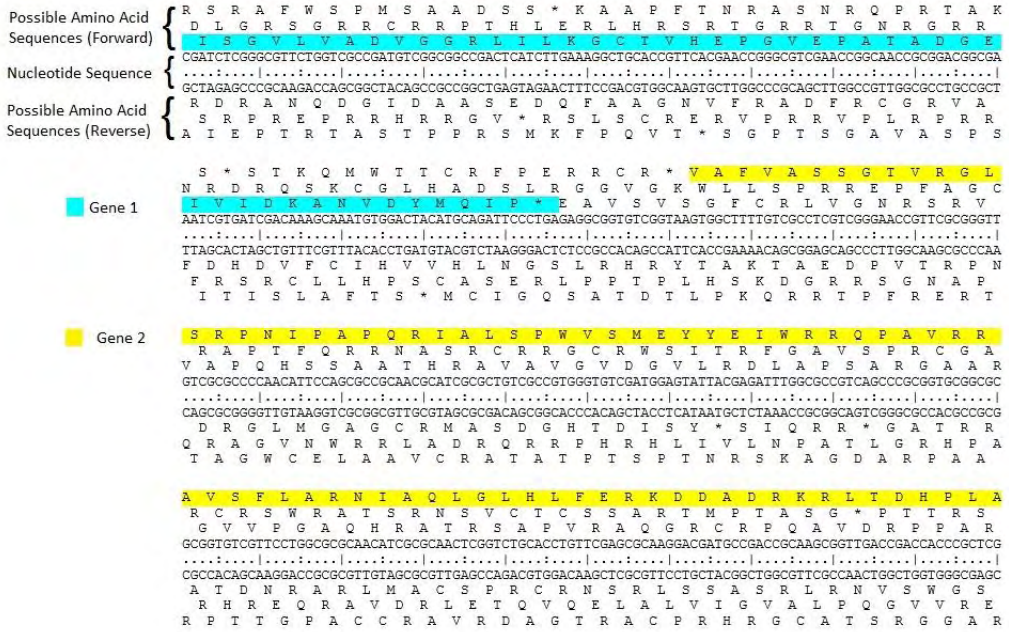


Figure 1.9: Six frame translations (Thatsonginc, 5-7-2014, Creative Commons)

1.10 Coding region

The coding region of a gene, also known as the coding sequence or CDS (from coding DNA sequence), is that portion of a gene's DNA or RNA, composed of exons, that codes for protein. The region is bounded nearer the 5' end by a start codon and nearer the 3' end with a stop codon. The coding region in mRNA is bounded by the five prime untranslated region and the three prime untranslated region, which are also parts of the exons. The CDS is that portion of an mRNA transcript that is translated by a ribosome.

While identification of open reading frames within a DNA sequence is straightforward, identifying coding sequences is not, because the cell translates only a subset of all open reading frames to proteins. Currently CDS prediction uses sampling and sequencing of mRNA from cells, although there is still the problem of determining which parts of a given mRNA are actually translated to protein. CDS prediction is a subset of gene prediction, the latter also including prediction of DNA sequences that code not only for protein but also for other functional elements such as RNA genes and regulatory sequences.

Chapter 2
Introduction

2.1 Introduction to the problem

In February 2001, the first assembly of the human genome was published by the International Human Genome Sequencing Consortium (Consortium, 2001). According to this study human gene tend to have small exons (encoding in average only 50 codons) separated by long introns (some exceeding 10 kb). This phenomenon increases the signal-to-noise ratio for algorithms that attempt to facilitate gene prediction, leading to significantly limited accuracy. Instead, the performance of such algorithms relies mostly on the availability of coding sequences that can be utilized to develop robust predictive models. This thesis presents a novel in silico approach that can readily identify the coding segments of either known or putative genic loci. To this end, the algorithm must be capable of: i) locating all Open Reading Frames (ORFs) located in the queried sequence, ii) identifying the correct ORF, whose 5'end is considered the Translation Initiation Site (TIS). The ORF is defined as a stretch of DNA codons (triplets of nucleotides) that start with ATG (start codon) and terminate with TAA/TAG/TGA (stop codons). Stop codons are not considered part of the ORF. Since all codons represent triplets of nucleotides, there can be up to three ORFs per single stranded DNA sequence. The immediate flanking loci of coding regions do not encode for proteins and can be considered as non-coding (3' and 5' UnTranslated Regions, UTRs). This is a phenomenon that the proposed methodology attempts to exploit since in general, the patterns of nucleotide composition greatly differ between coding and non-coding sequences.

The original work for the identification of TIS in coding sequences dates back to 1987, when Kozak developed the first weight matrix from an extended collection of data (Kozak, 1987). The consensus motif derived from this matrix was GCCACCatgG, describing a G residue following the ATG codon, and a purine, preferably A, three nucleotides upstream, as two highly conserved positions that exert the strongest effect. While attempting to describe what really happens in the cell, Kozak developed the ribosome-scanning model. According to this model (Kozak, 1996), ribosomes initially attach to the specific cap region

in the 5' end of mRNAs and subsequently scan the sequence until they find the first ATG located in an optimal nucleotide context. This is described as the site where the translation of codons into amino acids begins. Although this process characterizes most studied mRNA's, there are some notable exceptions (Kozak, 1996 and Pain,1996).

In cases where the first ATG codon of the sequence has a less than optimal nucleotide context, it can actually be bypassed by the ribosome, which then initiates translation from a subsequent start codon located in a more optimal nucleotide context further downstream. This phenomenon is also known as leaky scanning.

In reinitiation, the translation starts from an ATG codon upstream of the coding region, located in optimal nucleotide context inside the 5' UTR region and is terminated at the first stop codon, normally in a short downstream distance. Scanning then continues until the authentic ATG codon (start codon) is reached.

In internal initiation, the ribosome directly binds near the authentic ATG codon. This is mostly a characteristic of some viral mRNAs of peculiar structure.

During the last decade, a plethora of computational methods has emerged aiming to facilitate the distinction between coding and non-coding sequences. Even though the amount of available implementations is quite large, only a small fraction provides source code for stand-alone usage, while the vast majority can only be accessed through web servers of limited throughput capacity. The most notable implementations for TIS or Coding Potential (CP) prediction that are also discussed and tested in this thesis include CPAT, CPC, PLEK, PORTRAIT and OrfPredictor.

In the following chapter, the rationale behind the state-of-the-art is explained in details.

Chapter 3

Existing methodologies for TIS prediction

3.1 The Coding Potential Assessment Tool or CPAT

The Coding Potential Assessment Tool or CPAT (Wang, Park et al. 2013) is a tool which is used to distinguish coding from non-coding genes or transcripts. The tool does not use an alignment method and is built with four sequence features. The first feature finds the open reading frame size of a gene or transcript. The second finds the open reading frame coverage. For the third feature the Fickett TESTCODE statistic method is used. And the forth feature finds the hexamer for the gene or transcript.

CPAT developers claimed that their software gave the highest results compared with other state-of-the-art methods. They also claimed that their method is not only accurate but also fast. That enables the users to give big datasets for prediction.

CPAT has also developed a web interface which allows users to give their sequences for online prediction and receive the results instantly.

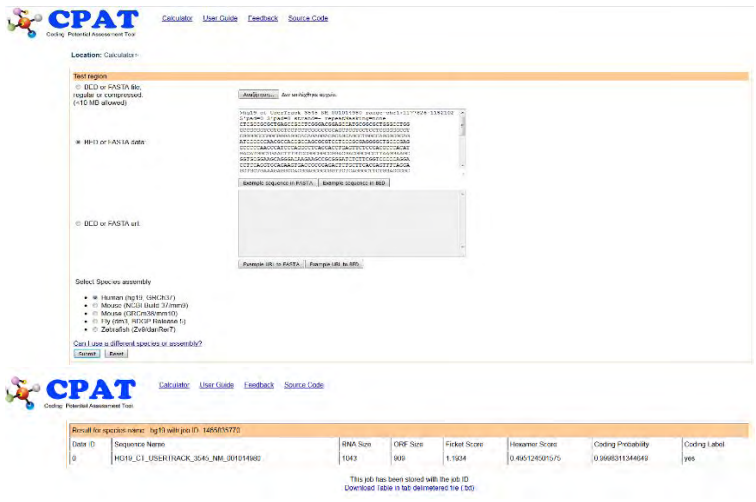


Figure 3.1: How CPAT works online. This figure has been designed for the purpose of this thesis.

3.2 Coding Potential Calculator or CPC

The Coding Potential Calculator or CPC(Kong, Zhang et al. 2007) is another method which is used to distinguish protein genes from non-coding. CPC is characterized by six sequence features. The three of the features are used to find the quality of Open Reading Frame (ORF) and the other three are used to give a score prediction of each sequence. These six features are given to a support vector machine (SVM) for training. More specifically, CPC uses the LIBSVM (SVM) method for training.

CPC developers claimed that their method can distinguish coding from non-coding transcripts with high accuracy. They also claimed that their method is extremely fast in predicting different test sets. Also CPC works online in a web-based interface at the following link: http://cpc.cbi.pku.edu.cn. The online CPC method also displays the detailed sequence features with graphics and figures.

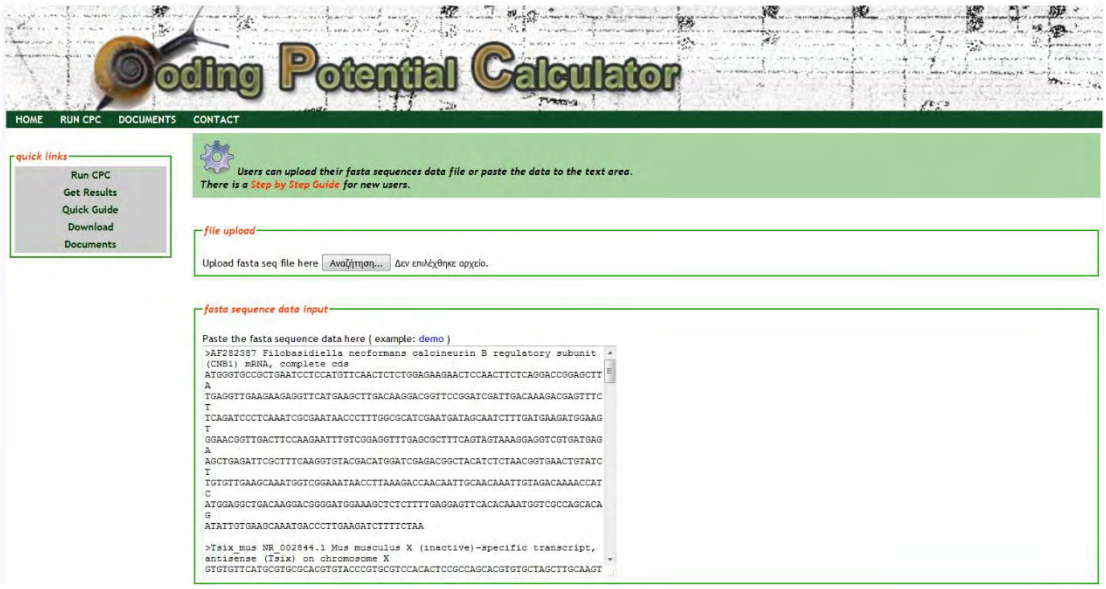


Figure 3.2: How CPC works for online prediction. This figure has been designed for the purpose of this thesis.

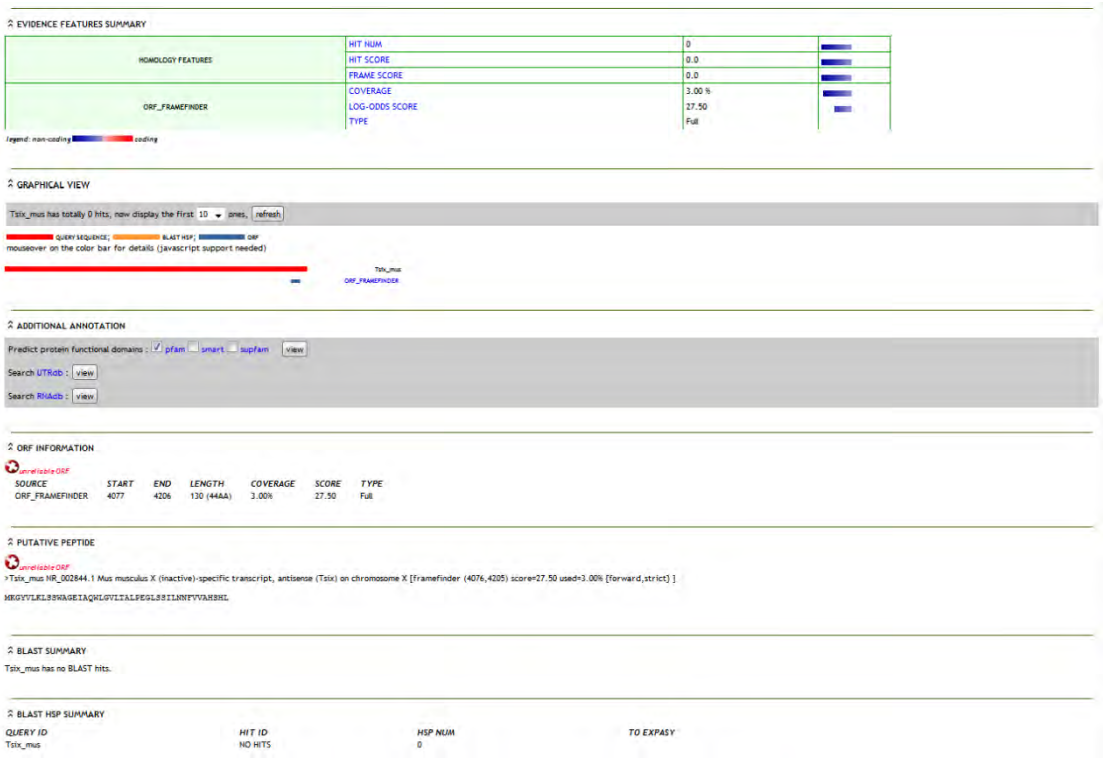


Figure 3.3: CPC online prediction example. This figure has been designed for the purpose of this thesis.

3.3 Predictor of long non-coding RNAs and messenger RNAs based on an improved k-mer scheme or PLEK

PLEK (Li, Zhang et al. 2014) is another tool which is used to distinguish mRNAs from non-coding RNAs. More specifically, mRNAs from lncRNAs. PLEK is an alignment free method, built from five k-mer binary features with sliding windows. The first feature is the 1-mer. The second feature is the 2-mer. The third and fourth features are 3-mer and 4-mer and the last feature is 5-mer. PLEK uses these binary features on a support vector machine (SVM) with LIBSVM for training.

PLEK authors claimed that their method is eightfold faster than a newly developed alignment-free tool. And extremely faster than one of the most popular alignment-based tool, Coding Potential Calculator (CPC). PLEK is only for stand-alone usage and can be downloaded from the link: http://sourceforge.net/projects/plek/files/latest/download.

3.4 Prediction of transcriptomic ncRNA by ab initio methods or PORTRAIT

PORTRAIT (Arrial, Togawa et al. 2009) is an algorithm which is suitable for the distinction between ncRNAs and mRNAs. PORTRAIT uses "ANGLE" software package to produce the features which have been trained with the support vector machine (SVM) method with LIBSVM package.

PORTRAIT is used only for stand-alone installation. The users can download the software from the following link: http://bioinformatics.cenargen.embrapa.br/portrait/download/.

3.5 OrfPredictor: predicting protein-coding regions in EST-derived sequences

OrfPredictor(Min, Butler et al. 2005) is the last method used in our thesis for the distinction of coding transcripts from non-coding. OrfPredictor is an online tool. The program uses BLASTX to distinguish coding and non-coding genes.

OrfPredictor is available from the link: https://fungalgenome.concordia.ca/tools/OrfPredictor.html. Also the users can ask the authors for the open-source software which is only for "Linux".

Chapter 4
Datasets

4.1 Positive/Negative set selection

Aim of the project described in this thesis was to develop a computational method able to distinguish between coding and non-coding DNA sequences. In order to build a robust predictive model, high quality and well annotated DNA sequences are required. This set of sequences needs to include positive (coding) as well as negative (non-coding) examples. The one will be used as a set of positive examples and the other one as a set of negative examples. For this reason, we downloaded from Ensembl (Yates, Akanni et al. 2016) human coding DNA transcripts and non-coding which are located upstream of cDNA. For additional information regarding the sets visit the Appendix.

The positive set consists of 11.565 well annotated coding DNA sequences and the negative set consists of 11.565 non-coding sequences which are located upstream of cDNA. This set, called "Initial set".

For the training of our method which called D-TIS used two sets. A training set which used for the training of the first part of D-TIS and another training set which used for the training of the second part of D-TIS.

The first training set consists of 6.666 positive and 6.666 negative sequences out of 11.565 of Initial set. The first training set, called "Training Set A".

The second training set consists of 3.334 positive and 3.334 negative sequences out of 11.565 of Initial set. The second training set, called "Training Set B".

For the test of D-TIS used two test sets. The first test set consists of 1.565 positive and 1565 negative sequences out of 11.565 of initial set. This test set, called "Test Set A".

The second test set built from 364 sORFs which derived from the supplementary material of (Mackowiak, Zauber et al. 2015) and 364 non-coding sequences from the upstream flank of cDNA. This set, called "Test

Set B". We created the Test Set B, because the existed TIS algorithms find difficult to distinguish the small Open Reading Frame sequences.

The "Initial set" downloaded by Ensembl which is an authorized and a reliable genome browser. Also D-TIS and the other TIS tools which described in chapter 3 except PORTRAIT and OrfPredictor tested on the Test Set A and Test Set B. PORTRAIT and OrfPredictor do not return the probability score for the predicted sequence so we could not to compare them with the other tools. For additional information regarding the sets visit the Appendix.

Chapter 5
Description of D-TIS algorithm

5.1 Algorithm Description

Aim of D-TIS algorithm is to go beyond the state-of-the-art in the scientific research topic of exploring the coding potential of genic DNA loci. To this end, two distinct features, named Ribosome Signal (RS) and Coding Potential (CP), have been utilized to train an equal number of Machine Learning models on a high-quality set of already annotated protein-coding and non-coding DNA sequences. In order to maximize the algorithm's performance, the results of both models are combined into a third one which provides the final predictions.

The one sensitive to the coding or non-coding potential around the start codon of a sequence, called Ribosome Signal and the other sensitive to the conserved motif, called Coding Potential. These two features are based on Graphic Process Unit Library for Support Vector Machines (Athanasopoulos, Dimou et al, 2011).

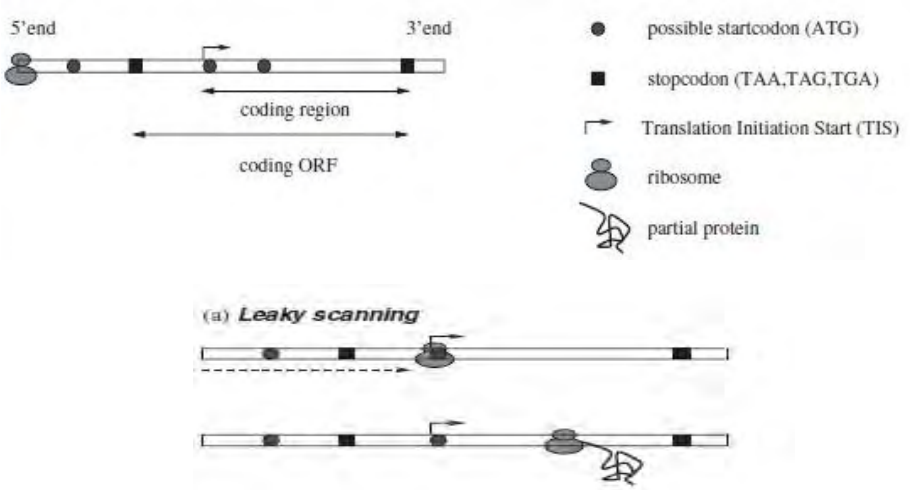
The rationale behind the choice of these two features is based on observations made by previous studies that highlight them as the most informative in terms of distinguishing between coding and non-coding sequences. However, D-TIS algorithm is the first computational method that integrates these two features into a robust Machine Learning framework.

In the following sections, the algorithmic procedure of D-TIS will be presented step by step.

5.1.1 Ribosome Signal Module

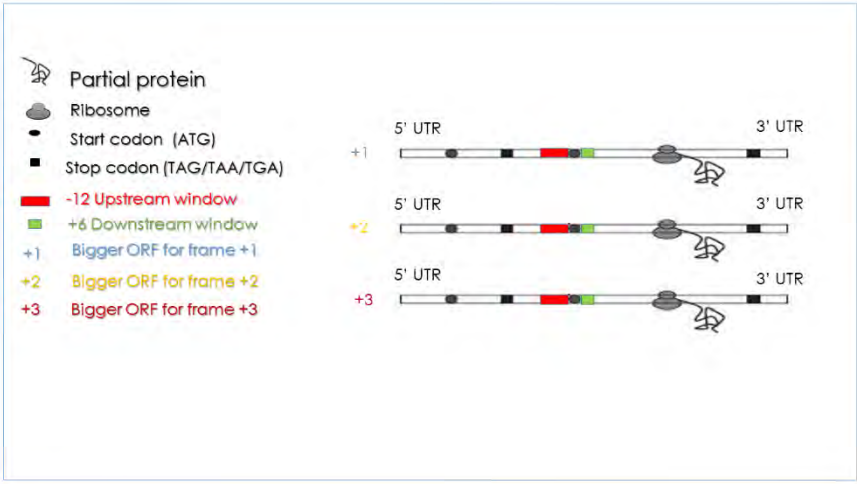
Prior to both feature calculations, for each queried sequence the algorithm identifies the largest Open Reading Frame (ORF) in the sense strand. An ORF is defined as a continuous stretch of codons (triplets of nucleotides) that start with ATG (start codon) and finish with TAA/TAG/TGA (stop codon). The stop codon is not included into the ORF however.

Subsequently, in order to calculate the RS feature, a window of 18 nucleotides long is applied. The window spans the positions from -12 to +9 surrounding the start codon (+1).



Figures 5.1 & 5.2: Graphical representation of the Translation Initiation Start (TIS) site by the ribosome, a procedure that leads to protein production. This figure has been derived from the study of (Hatzigeorgiou 2002).

The resulting 18 nucleotides are converted into a binary string based on a 4-digit code (figure 5.4). The binary string is directly forwarded to its corresponding Support Vector Machine (SVM) model in order to calculate the RS score of the queried sequence. The training process of the SVM models is presented in the following paragraphs of the current section.



Figures 5.3: How the Ribosome Signal picks the bigger ORF for each frame for a sequence. This figure has been designed for the purpose of this thesis.

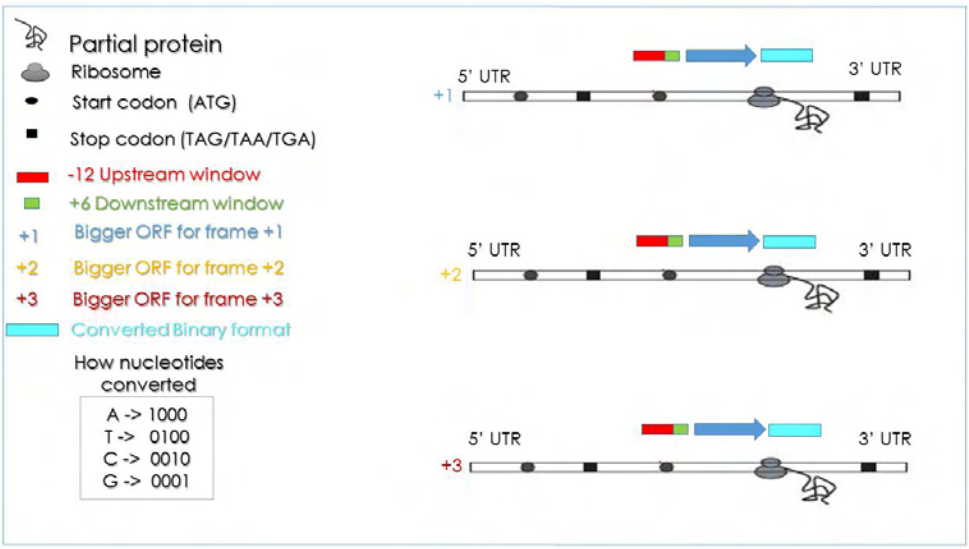


Figure 5.4: How Classifier Ribosome Signal converts the 18 nucleotides scanning window to a binary format. This figure has been designed for the purpose of this thesis.

Nucleotide	Binary Format
A	1 0 0 0
T	0 1 0 0
C	0 0 1 0
G	0 0 0 1

Table 5.1: Binary format for Ribosome Signal. This table has been designed for the purpose of this thesis.

The RS SVM model was trained on positive and negative instances of open reading frame start codons. The positive set was Training set A. The negative set was derived by randomly selecting one of the remaining ORFs that belonged to each coding sequence in the positive

set. The ratio between positive and negative instances was 1:1. In order to significantly speed up both the training and test process, a modified version of libsvm (GPU LIBSVM) was utilized which is based on CUDA programming and exploits the multicore capabilities of a computer's graphics processing unit (Athanasopoulos, Dimou et al, 2011). This resulted in processing a model of 13.332 instances in 15 minutes as compared to 6 hours in a typical multi-CPU environment.

The optimal values of C (log2c=1) and gamma (log2g=-3) parameters of the SVM model were found by applying a grid search approach which resulted in 78.70% accuracy. The best kernel function for our training set was Radial Basis Function (RBF). The "5-fold" cross-validation accuracy of the model was 78%. An example of the model's input and output is presented in Tables 5.2 and 5.3.

Nucleotide Sequence	Binary Format
CCGGAG	001000100001000110000001

Table 5.2: Example, conversion of Nucleotide Sequence to Binary format for Ribosome Signal. This table has been designed for the purpose of this thesis.

Label	Positive probability Score	Negative probability Score
1	0.6884	0.3116

Table 5.3: Example, what the SVM returns for a single gene. This table has been designed for the purpose of this thesis.

For the distinction of our training set we used the positive probability score and the real label given by us for each set. Plus, for the elements of a positive set and minus for the elements of a negative set (+1, -1).

5.1.2 Coding Potential Module

The CP module of D-TIS utilizes a scanning window that starts 60 nucleotides downstream of the start codon. Four windows of different length were utilized in order to evaluate whether the size parameter influences the model's performance in recognizing the coding region of gene loci.

The first scanning window initiates at the 60th nt downstream of each start codon and has a size of 60 nts. Therefore, this window was named Classifier Coding Potential +60 - +120 (Figure 5.4).

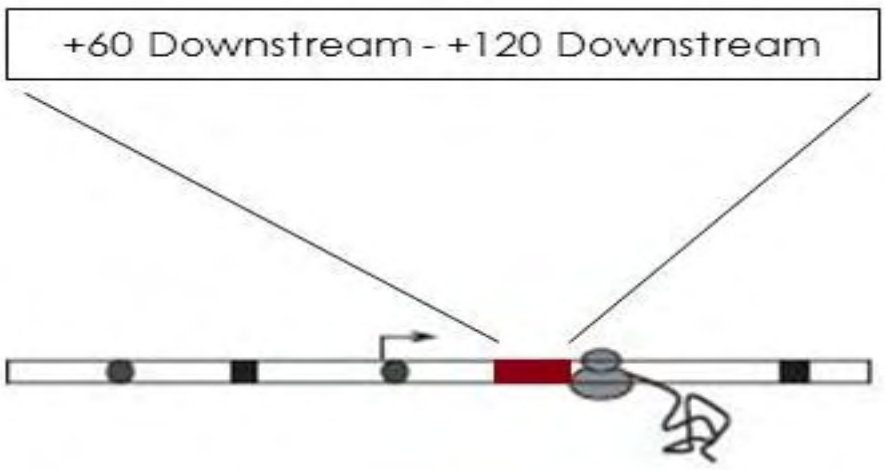


Figure 5.5: How Classifier Coding Potential create +60 - +120 downstream scanning window. This figure has been designed for the purpose of this thesis.

The second scanning window also initiates at the 60th nucleotide downstream of each start codon and its size is 90nts. Therefore, it is named Classifier Coding Potential +60 - +150 (Figure 5.5).

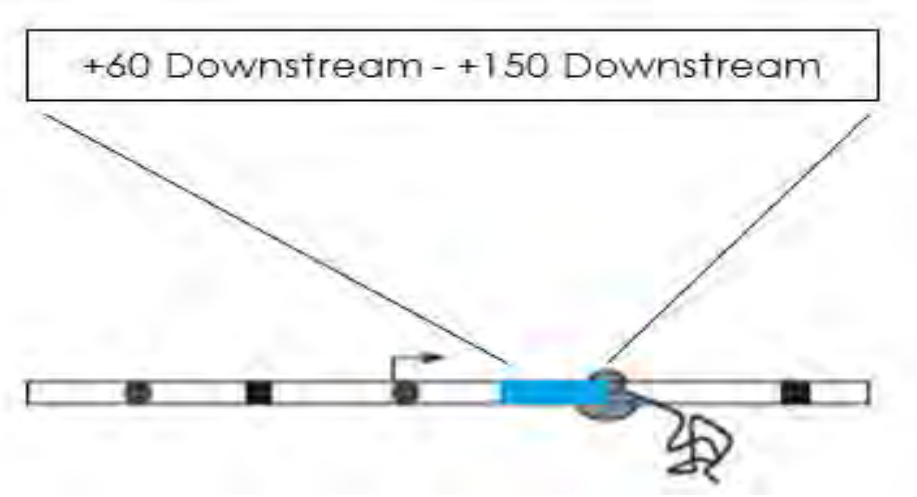


Figure 5.6: How Classifier Coding Potent Potential create +60 - +150 downstream scanning window. This figure has been designed for the purpose of this thesis

The third scanning window initiates at the 60th nucleotide downstream of each start codon and its size is 120nts. Therefore, it is named Classifier Coding Potential +60 - +180 (Figure 5.7).

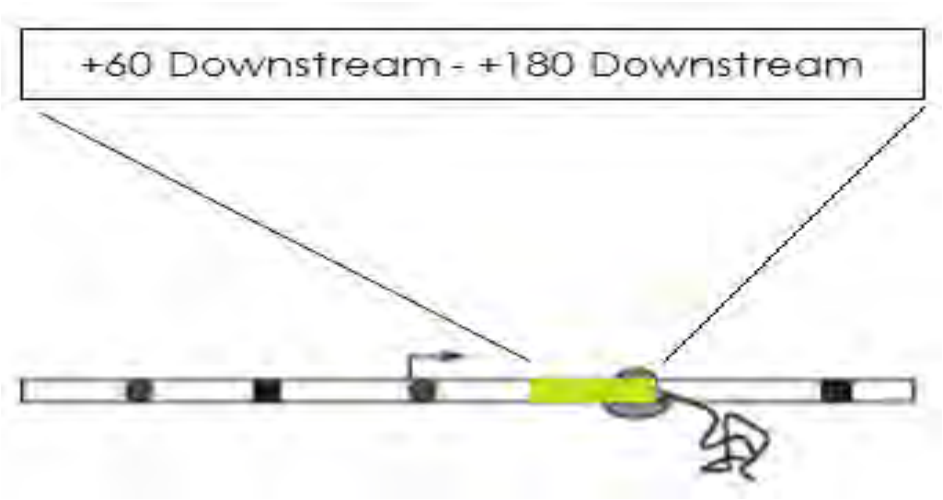


Figure 5.7: How Classifier Coding Potential create +60 - +180 downstream scanning window. This figure has been designed for the purpose of this thesis.

The final scanning window also initiates at the 60th nucleotide downstream of each start codon and its size is 150nts. Therefore, it is named *Classifier Coding Potential +60 - +210* (Figure 5.8).

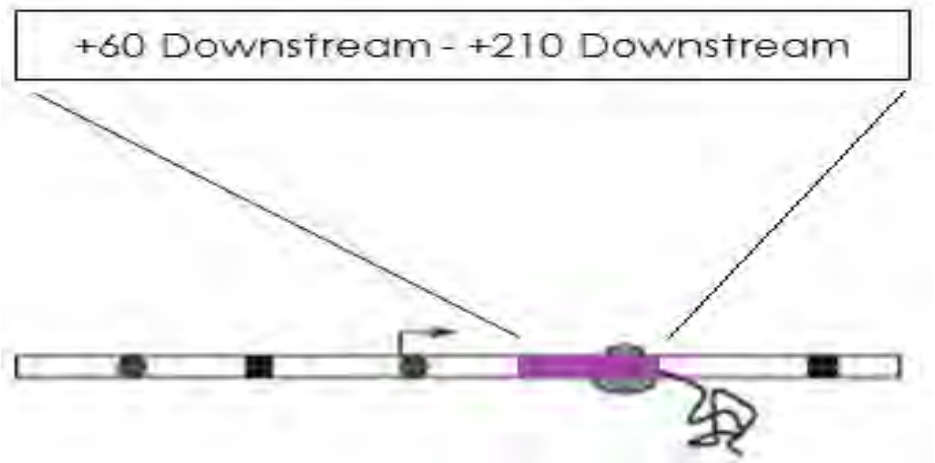


Figure 5.8: How Classifier Coding Potential create +60 - +210 downstream scanning window. This figure has been designed for the purpose of this thesis.

The highest results for the recognition of the coding region returned from the Classifier Coding Potential +60 - +180.

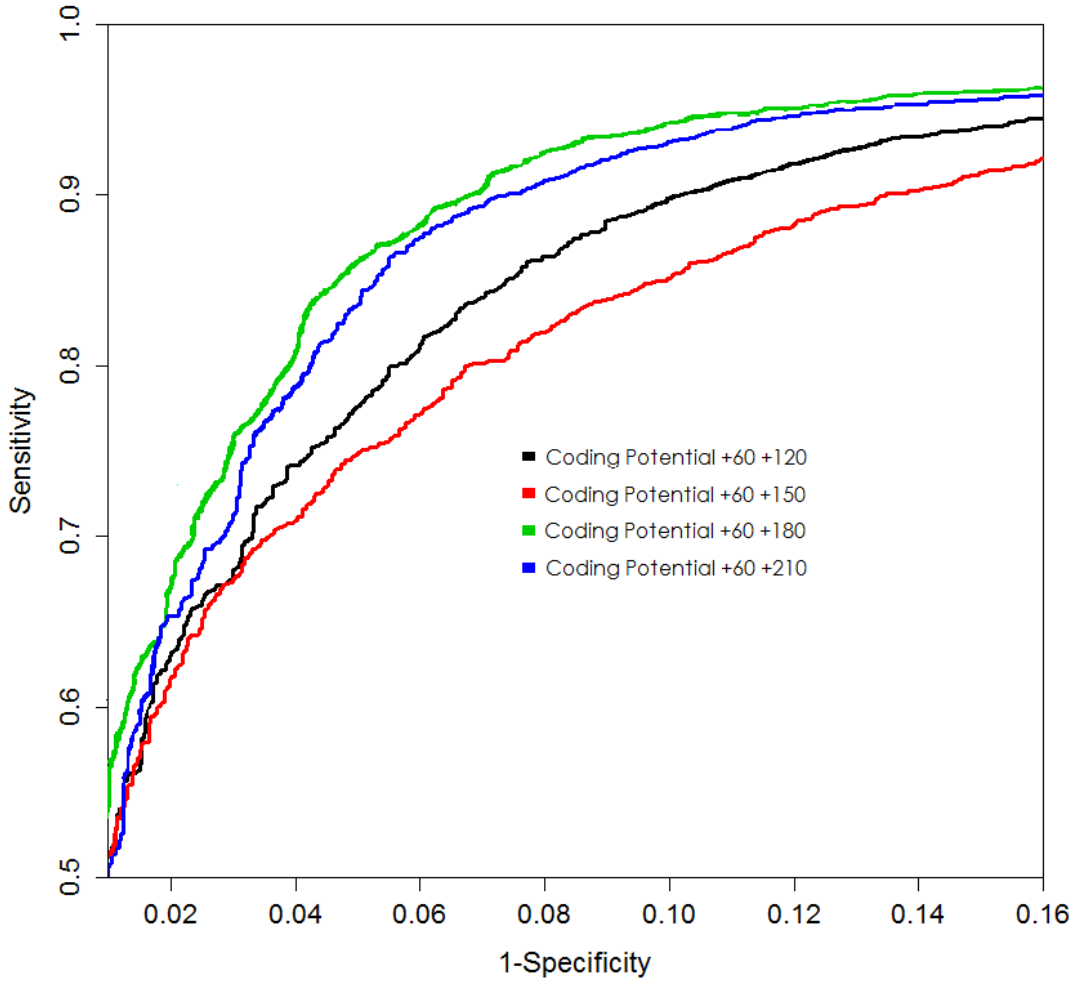


Figure 5.9: Results for the different windows. This figure has been designed for the purpose of this thesis.

Windows	AUC
Coding Potential +60 +120	0.9615
Coding Potential +60 +150	0.9517
Coding Potential +60 +180	0.9703
Coding Potential +60 +210	0.9669

Table 5.4: The AUC scores from sliding windows for Test Set A. This table has been designed for the purpose of this thesis.

The evaluation process of each window was based on Training set A which consists of 6.666 well annotated coding DNA sequences (positive set) and 6.666 non-coding sequences (negative set) located upstream of the positive instances. The results indicate that the best

performing window was Coding Potential +60 - +180 (Figure 5.9) which was incorporated into the D-TIS algorithm.

As described the Coding Potential as Ribosome Signal keeps the bigger ORF for each sequence.

The CP module of the algorithm utilizes the best performing window by applying a sliding strategy. It initiates 4 nts downstream of the start codon and the process is terminated at the last nucleotide before the in-frame stop codon of the ORF. Subsequently, every sliding window is converted into a binary string based on the previously described 4-digit code. The resulting binary string is forwarded to the CP SVM model which assigns a score to the corresponding queried sequence. The final score of the CP module is the average SVM score for each scanning window.

$$\text{Sliding Window} = \frac{\text{ORF sequence} - 120}{3}$$

Figure 5.10: How we calculated the sliding window for Coding Potential +60-+180. This figure has been designed for the purpose of this thesis.

The optimal parameters on GPU LIBSVM method were found to be C = 8 and gamma= 0.0078125 with an accuracy of 82.37%, as determined by grid search. The best kernel function for our training set was Radial Basis Function (RBF). The "5-fold" cross-validation accuracy of the model was 82%. An example of the model's output is presented in Table 5.6.

An example of the CP module input and output is presented in Tables 5.5 and 5.6.

Nucleotide Sequence	Binary Format
CCGGAG	001000100001000110000001

Table 5.5: Example, how converted Nucleotide Sequence to Binary format for Coding Potential +60+180. This table has been designed for the purpose of this thesis.

Label	Positive probability Score	Negative probability Score
1	0.6884	0.3116

Table 5.6: Example, what the SVM return for a single gene. This figure has been designed for the purpose of this thesis.

5.1.3 Extended Classifier

The Extended Classifier is the final module of D-TIS. It combines the results of RS and CP module in order to extract the final score of each queried sequence. This module was trained on Training Set B which consists of 3.334 coding DNA sequences (positive set) and 3.334 non-coding sequences (negative set) derived from the immediately upstream regions of the coding sequences in the positive set. An overview of the algorithm is presented in Figure 5.10.

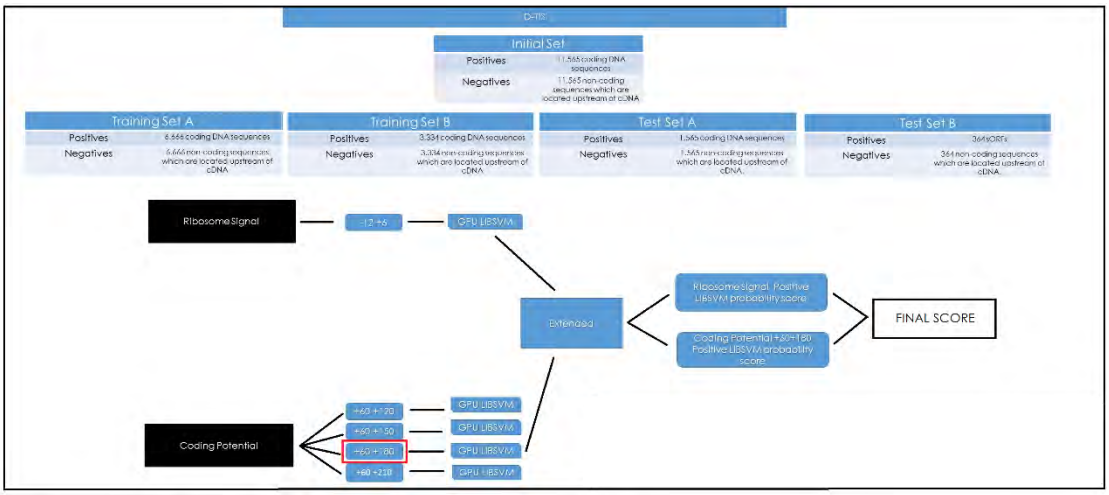


Figure 5.11: How D-TIS works. This figure has been designed for the purpose of this thesis.

The optimal values of C (13, log2 scale) and gamma (1, log2 scale) parameters were found by applying a grid search approach which reported 93.07% accuracy. The best kernel function for our training set was Radial Basis Function (RBF). The "5-fold" cross-validation accuracy of the model was 92%. An example of the model's output is presented in Table 5.8.

An example of the CP module input and output is presented in Tables 5.7 and 5.8.

Ribosome Signal Libsvm Positive Probability Score	Coding Potential Libsvm Positive Probability Score
0.7875	0.8987

Table 5.7: Example, what our method give to the SVM. This table has been designed for the purpose of this thesis.

Label	Positive probability Score	Negative probability Score

1	0.9644	0.0356

Table 5.8: Example, what the SVM return for a single gene. This table has been designed for the purpose of this thesis.

The Test set A and B were used for the Extended Classifier and returned the final score for D-TIS. As described above, the Test set A was built from 1.565 coding DNA sequences as the positive set and 1.565 non-coding sequences from the upstream flank of cDNA as the negative set. The Test set B was built from 354 sORFs and 354 non-coding sequences from the upstream flank of cDNA. The results from Test set A and the Test set B were the final results of D-TIS and were used for the comparison among the other coding potential algorithms, like CPAT, CPC and PLEK.

At this point, it must be mentioned that D-TIS compared CPAT, CPC and PLEK algorithms and the reason was that PORTRAIT and OrfPredictor did not return a probability score for the given test set. They only returned a label which is used to show if the gene given was positive, when the label format was equal to 1 or negative, when the label format was equal to -1.

The comparison of our method with PORTRAIT and OrfPredictor could only be established, if the default threshold used for comparison in machine learning was equal to 0.5 - something that will happen in the future.

Chapter 6
Comparing the Results

6.1 Comparing the Results for the Test set A

The purpose of this section is to present the comparison for the D-TIS versus CPAT, CPC and PLEK for the Test set A.

As described above, D-TIS could only be compared with CPAT CPC and PLEK. The reason was that PORTRAIT and OrfPredictor do not give the probability score for the test sets, so rejected. For the comparison we used the Test set A. The Test set A consisted of 1.565 positive sequences and 1.565 negative sequences out of 11.565 sequences of the Initial set.

To compare D-TIS versus CPAT, CPC and PLEK we used different types of diagrams, such as Receiver Operating Characteristic curve, or more commonly known as "ROC" curve, "Precision-Recall" and "Recall-Precision" curves, "True Positives-True Negatives" curve and finally "False Positives-False Negatives" curve.

For our comparison among these methods the first thing we did was to use the probability positive scores which returned for the Test set A from each algorithm and to create a table with the final results for the different thresholds.

The table for each prediction of each algorithm consists of *Threshold, TP, FP, TN, FN, Precision, Recall, Specificity, Sensitivity and Accuracy*.

One last thing which must be mentioned is that the tables which were created to find the results above had different probability scores and threshold. Only D-TIS and CPAT had the same probability range which was "0 – 1". CPC and PLEK used their own probability ranges.

After that, we took the results from each table for each algorithm and created the different diagrams mentioned above to make the comparison between the algorithms.

6.1.1 ROC Curve Comparison for Test Set A

The Receiver Operating Characteristic curve or more commonly known as ROC Curve was the first diagram used for the comparison between the algorithms.

The ROC Curve is a graphical plot that illustrates the performance of a classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate known as (TPR) against the false positive rate known as (FPR) at various threshold settings. The true-positive rate is also known as "Sensitivity" or "recall" in machine learning. The false-positive rate is also known as the "Specificity" and can be calculated as "1 – Specificity". The ROC Curve created by Sensitivity for y-axis and Specificity for x-axis for each threshold and all thresholds create the area under the curve, known as "AUC".

The mathematical formula used to calculate the "Sensitivity" and the "Specificity" is given in the figures below:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \qquad \text{Specificity} = \frac{TN}{FP + TN}$$

Figures 6.1 & 6.2: Mathematical formula for Sensitivity and Specificity. This figure has been designed for the purpose of this thesis.

The "ROC" curve for D-TIS versus CPAT, CPC and PLEK is shown in the figure below:

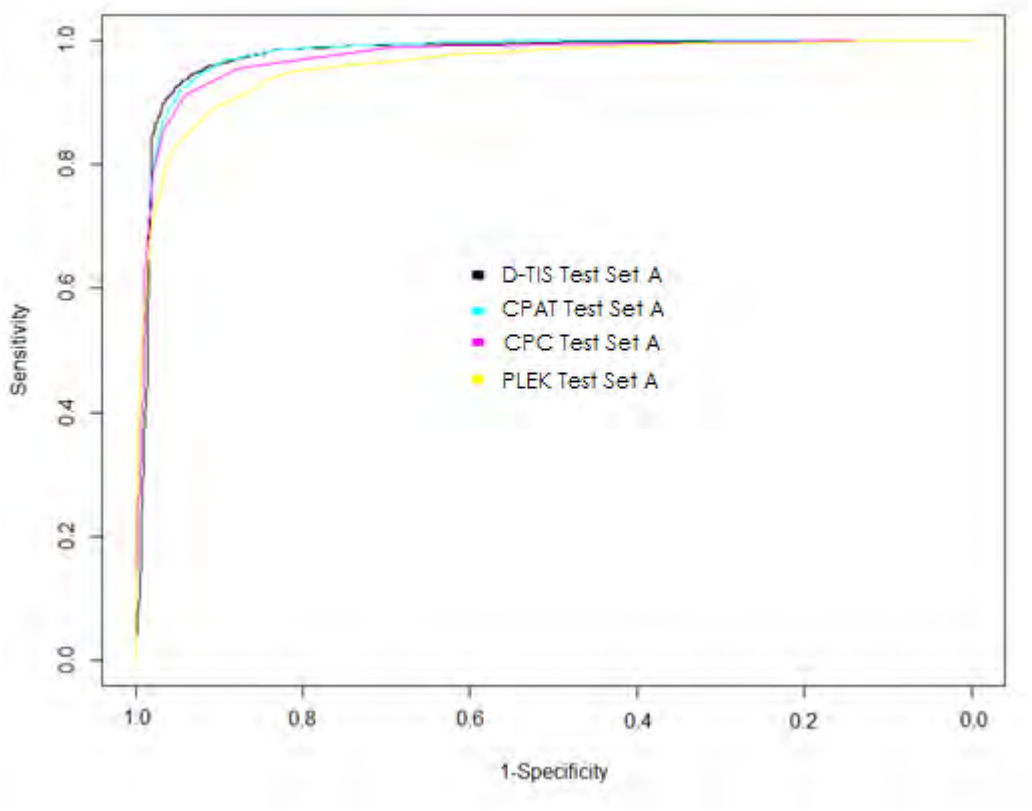


Figure 6.3: ROC Curve for Test Set A. This figure has been designed for the purpose of this thesis.

As shown in (figure 6.3) D-TIS with CPAT gave the highest "AUC" score. After that, CPC followed and last PLEK. The area under the curve (AUC) for each program is given in the table below:

ALGORITHM	AUC
D-TIS	0.9766
CPAT	0.9747
CPC	0.9738
PLEK	0.9589

Table 6.1: The AUC scores from algorithms for Test SET A. This table has been designed for the purpose of this thesis.

The "ROC" curve is one of the best methods to compare algorithms based on machine learning. To give more reliable results we need to take into consideration other diagrams too, especially "Precision-Recall".

6.1.2 Precision-Recall Curve for Test Set A

In pattern recognition and information retrieval with binary classification "Precision" is also called positive predictive value. It is the fraction of retrieved instances that are relevant, while "Recall", also known as "sensitivity", is the fraction of relevant instances that are retrieved. Both "Precision" and "Recall" are therefore based on an understanding and measure of relevance.

The mathematical formula which is used to calculate the "Precision" and "Recall" is given in the figures below:

$$\text{Precision} = \frac{TP}{TP + FP} \qquad \text{Recall} = \frac{TP}{TP + FN}$$

Figures 6.4 & 6.5: Mathematical formula for Precision and Recall. This figure has been designed for the purpose of this thesis.

The "Precision – Recall" and "Recall – Precision" curves for D-TIS versus CPAT, CPC and PLEK are shown in the figures below:

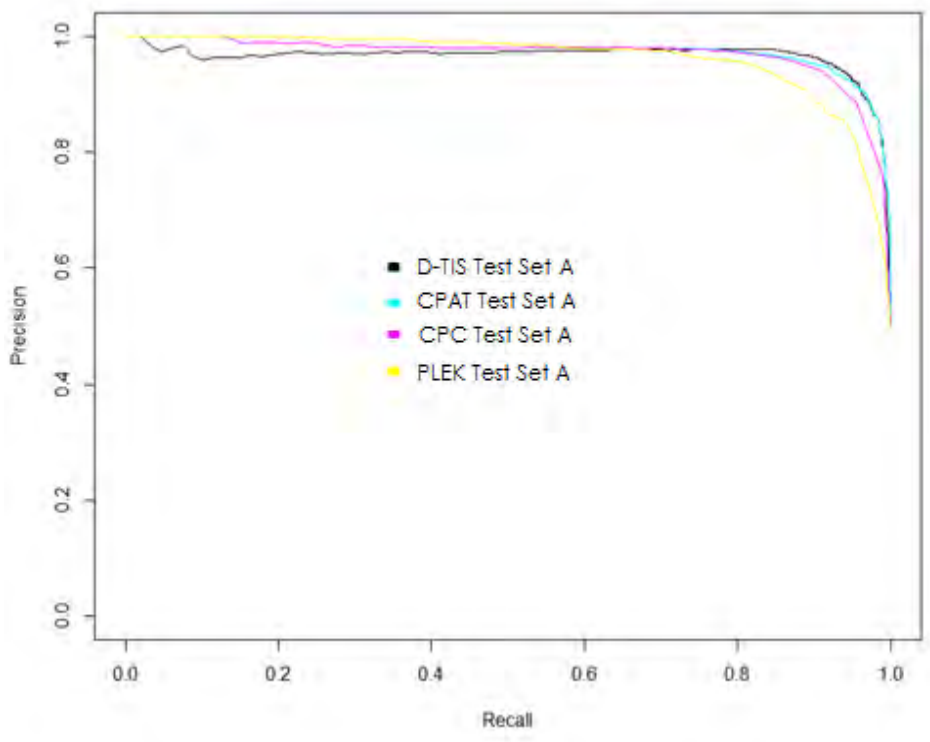


Figure 6.6: Precision - Recall curve for Test Set A. This figure has been designed for the purpose of this thesis.

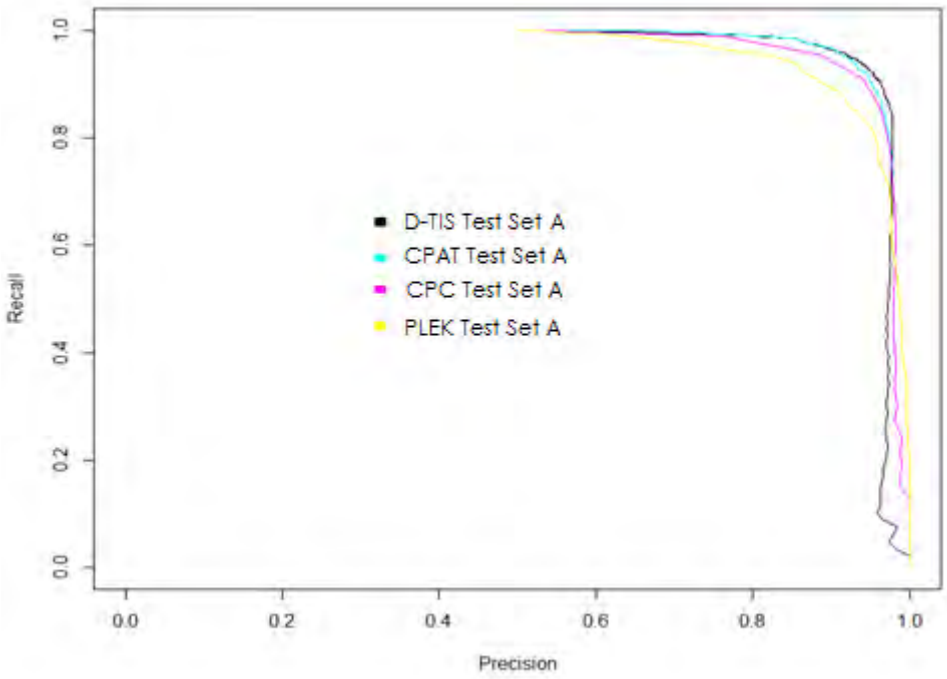


Figure 6.7: Recall Recall – Precision curve for Test Set A. This figure has been designed for the purpose of this thesis.

As shown in the diagrams above, D-TIS and CPAT returned the highest results and then came CPC and PLEK. Our method gave high "Precision" with high "Recall".

6.1.3 TP-TN Curve for Test Set A

The "TP-TN" curve is a diagram which consists of the "True Positive" in y-axis and the "True Negative" in x-axis. The range of the two axis consist of the range of Test set A. So, the range of the Test set A was "0-1565". This diagram can calculate how many "True Positives" and "True Negatives" an algorithm can predict for the different thresholds. "TP-TN" curves are shown in the figure below:

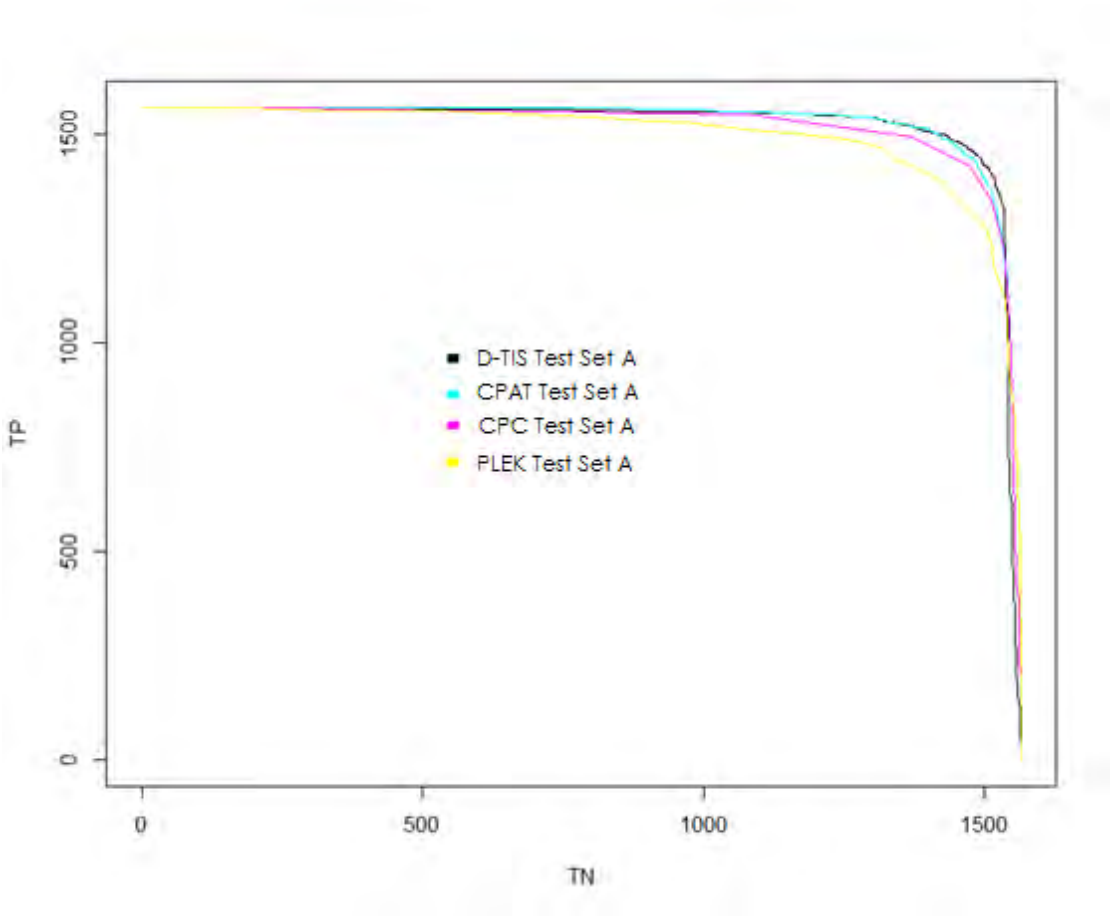


Figure 6.8: TP – TN curve for Test Set A. This figure has been designed for the purpose of this thesis.

In this diagram D-TIS and CPAT returned the best results and as in the previous diagrams CPC and PLEK came third and fourth.

6.1.4 FP-FN Curve for Test Set A

The fourth and last diagram used to calculate D-TIS versus CPAT, CPC and PLEK was the "False Positive" and "False Negative" diagram.

The y-axis consists of "False Positives" and the x-axis from "False Negatives". The range of axis was the range of the Test set A. So the range was "0-1565". The purpose of this diagram was to show which algorithm can predict "True Positives" and "True Negatives" with high accuracy for the different ranges of threshold. See the diagram in the figure below:

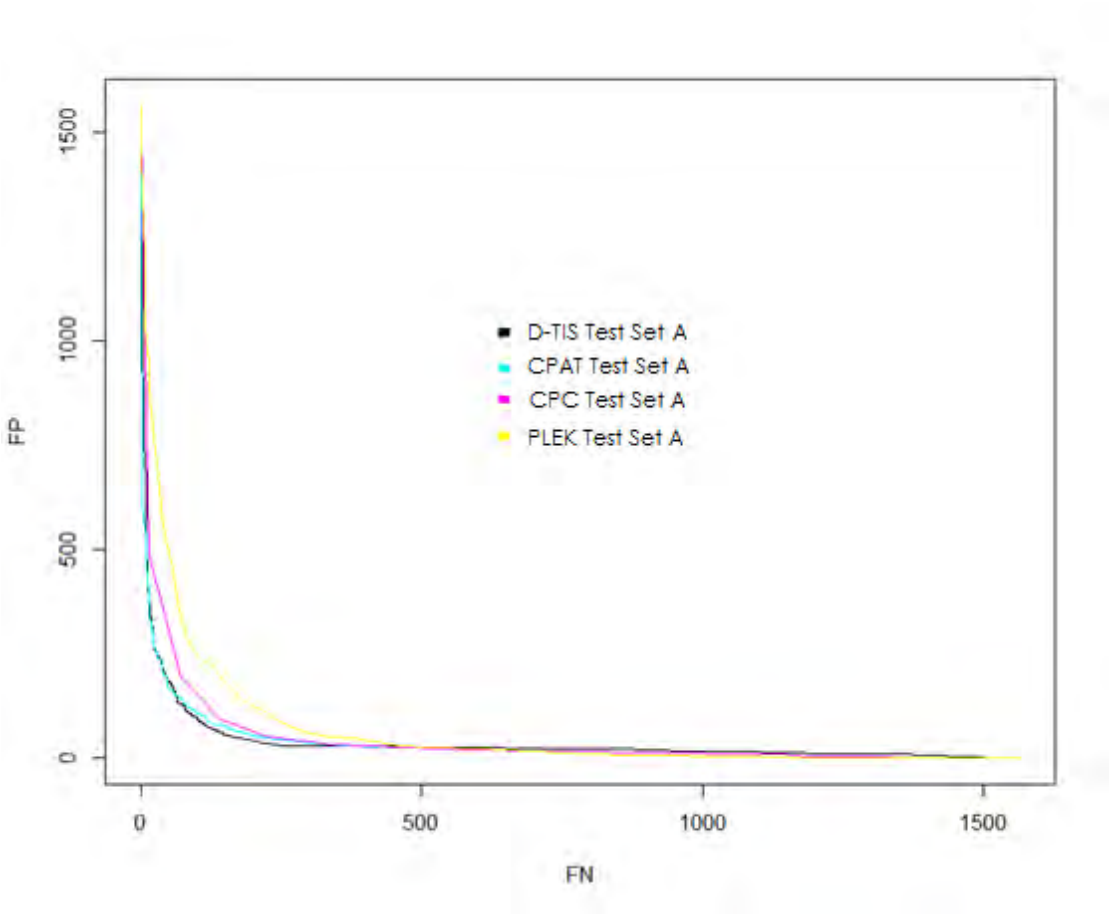


Figure 6.9: FP – FN curve for Test set A. D-TIS with CPAT returned the best results and after came CPC and last PLEK. This figure has been designed for the purpose of this thesis.

6.1.5 Thresholds Final Results for Test Set A

The diagrams showed that D-TIS returned the best results for the prediction of the Test set A. After it, CPAT followed, while CPC and PLEK came third and fourth respectively.

To be more accurate, one must choose a specific threshold for the prediction of the Test set A or to present the behavior of all the algorithms for the different thresholds. The tables below returned for each threshold show how accurate the algorithms were. As mentioned before, D-TIS and CPAT had threshold range "0-1", but CPC and PLEK had their own ranges. For more details, check the tables:

D-TIS threshold table for Test Set A:

Threshold	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
0.1	1542	292	1273	23	0.8407	0.9853	0.8134	0.9853	0.8993
0.2	1523	207	1358	42	0.8803	0.9731	0.8677	0.9731	0.9204
0.3	1507	163	1402	58	0.9023	0.9629	0.8958	0.9629	0.9293
0.4	1495	130	1435	70	0.92	0.9552	0.9169	0.9552	0.9361
0.5	1486	119	1446	79	0.9258	0.9495	0.9239	0.9495	0.9367
0.6	1468	97	1468	97	0.9380	0.9380	0.9380	0.9380	0.9380
0.7	1447	74	1491	118	0.9513	0.9246	0.9527	0.9246	0.9386
0.8	1420	59	1506	145	0.9601	0.9073	0.9623	0.9073	0.9348
0.9	1355	37	1528	210	0.9734	0.8658	0.9763	0.8658	0.9210
1.0	2	0	1565	1563	1.0	0.0012	1.0	0.0012	0.5006

Table 6.2: D-TIS thresholds table for Test Set A. This table has been designed for the purpose of this thesis.

CPAT thresholds table for Test Set A:

Threshold	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
0.1	1560	573	992	5	0.7313	0.9968	0.6338	0.9968	0.8153
0.2	1551	404	1161	14	0.7933	0.9910	0.7418	0.9910	0.8664
0.3	1546	320	1245	19	0.8285	0.9878	0.7955	0.9878	0.8916
0.4	1541	267	1298	24	0.8523	0.9846	0.8293	0.9846	0.9070
0.5	1527	225	1340	38	0.8715	0.9757	0.8562	0.9757	0.9159
0.6	1517	178	1387	48	0.8949	0.9693	0.8862	0.9693	0.9277
0.7	1503	149	1416	62	0.9098	0.9603	0.9047	0.9603	0.9325
0.8	1482	121	1444	83	0.9245	0.9469	0.9226	0.9469	0.9348
0.9	1447	90	1475	118	0.9414	0.9246	0.9424	0.9246	0.9335
1.0	1140	24	1541	425	0.9793	0.7284	0.9846	0.7284	0.8565

Table 6.3: CPAT thresholds table for Test Set A. This table has been designed for the purpose of this thesis.

CPC thresholds table for Test Set A:

Threshold	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
-0.831893	1549	488	1077	16	0.7604	0.9897	0.6881	0.9897	0.8389
-0.224236	1423	91	1474	142	0.9398	0.9092	0.9418	0.9092	0.9255
0.383421	1223	31	1534	342	0.9752	0.7814	0.9801	0.7814	0.8808
0.991078	974	18	1547	591	0.9818	0.6223	0.9884	0.6223	0.8054

1.598735	780	16	1549	785	0.9798	0.4984	0.9897	0.4984	0.7440
2.206392	596	11	1554	969	0.9818	0.3808	0.9929	0.3808	0.6869
2.814049	467	8	1557	1098	0.9831	0.2984	0.9948	0.2984	0.6466
3.421706	381	4	1561	1184	0.9896	0.2434	0.9974	0.2434	0.6204
4.3331915	266	3	1562	1299	0.9888	0.1699	0.9980	0.1699	0.5840
16.182503	4	0	1565	1561	1	0.0025	1	0.0025	0.5012

Table 6.4: CPC thresholds table for Test Set A. This table has been designed for the purpose of this thesis.

PLEK thresholds table for Test Set A:

Threshold	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
-1.599223	1563	1363	202	2	0.5341	0.9987	0.1290	0.9987	0.5638
-1.127835	1557	1080	485	8	0.5904	0.9948	0.3099	0.9948	0.6523
0.28633	1388	140	1425	177	0.9083	0.8869	0.9105	0.8869	0.8987
0.488353	1322	91	1474	243	0.9355	0.8447	0.9418	0.8447	0.8932
0.690377	1231	52	1513	334	0.9594	0.7865	0.9667	0.7865	0.8766
0.959742	1048	25	1540	517	0.9767	0.6696	0.9840	0.6696	0.8268
1.161765	871	15	1550	694	0.9830	0.5565	0.9904	0.5565	0.7734
1.498471	606	5	1560	959	0.9918	0.3872	0.9968	0.3872	0.6920
1.902518	359	1	1564	1206	0.9972	0.2293	0.9993	0.2293	0.6143
3.586048	2	0	1565	1563	1	0.0012	1	0.0012	0.5006

Table 6.5: PLEK thresholds table for Test Set A. This table has been designed for the purpose of this thesis.

The results obtained from the thresholds tables showed that D-TIS gave the highest accuracy. The type of accuracy is:

$$\text{Accuracy} = \frac{TP + TN}{P + N}$$

Figure 6.10: Mathematical formula for Accuracy. This figure has been designed for the purpose of this thesis.

A good method to finish with our comparison of the Test set A was to set as a threshold "0.5" which is the default threshold for machine learning.

In the table below we can see the comparison between algorithms for set threshold equal to "0.5". For CPC and PLEK we used the threshold which had the best results for each program. CPC threshold was equal to "- 0.224236" and PLEK equal to "0.28633".

Algorithm	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
D-TIS	1486	119	1446	79	0.9258	0.9495	0.9239	0.9495	**0.9367**
CPAT	1527	225	1340	38	0.8715	0.9757	0.8562	**0.9757**	0.9159
CPC	1423	91	1474	142	0.9398	0.9092	**0.9418**	0.9092	0.9255
PLEK	1388	140	1425	177	0.9083	0.8869	0.9105	0.8869	**0.8987**

Table 6.6: Algorithms final results for Test Set A. This table has been designed for the purpose of this thesis.

The table above shows that D-TIS had the highest "Accuracy" for the prediction of the Test set A. That means that D-TIS predicted with high rate the True Positives and the True Negatives and it predicted with low rate False Positives and False Negatives. That created a balance between "Specificity" and "Sensitivity".

CPAT had the highest "Sensitivity" but the lowest "Specificity" rate. So, CPAT predicted with high rate the True Positives but returned a lot of False Positives.

CPC had the highest "Specificity" but low "Sensitivity" rate. So, CPC predicted the True Negatives correctly but did not do so well with the True Positives as it returned a high rate of False Negatives.

PLEK had the lowest results. It predicted the True Negatives correctly but not the True Positives.

D-TIS, as shown in the different types of diagrams and the different sets of thresholds, predicted with high rate the True Positives and the True Negatives respectively. It kept a balance as an algorithm with high "Sensitivity", "Specificity" and "Precision" rates.

The other three methods were not so balanced in their results. CPAT, as described above, had an excellent rate for True Positives with a bad rate for True Negatives. Evenly, CPC had excellent "Specificity" rate but low "Sensitivity" rate. Finally, PLEK had the lowest "Accuracy" rate.

D-TIS is a reliable new method for coding potential. It is competitive compared to the other three methods, some of which are state-of-the art methods for coding potential, like CPAT and CPC.

6.2 Comparing the Results for Test Set B

The purpose of this section is to present the comparison for the D-TIS versus CPAT and CPC for the Test set B. PLEK was not included for the comparison between the D-TIS, CPAT and CPC. The reason was that PLEK do not designed for the prediction of small Open Reading Frames (sORFs).

For the comparison we used the Test set B. The Test set B consisted of 354 positive sORF sequences and 354 non-coding sequences from the upstream flank of cDNA.

To compare D-TIS versus CPAT and CPC we used different types of diagrams, such as Receiver Operating Characteristic curve, or more commonly known as "ROC" curve, "Precision-Recall" and "Recall-Precision" curves, "True Positives-True Negatives" curve and finally "False Positives-False Negatives" curve.

For our comparison among these methods the first thing we did was to use the probability positive scores which returned for the Test set B from each algorithm and to create a table with the final results for the different thresholds.

The table for each prediction of each algorithm consists of Threshold, TP, FP, TN, FN, Precision, Recall, Specificity, Sensitivity and Accuracy.

One last thing which must be mentioned is that the tables which were created to find the results above had different probability scores and threshold. Only D-TIS and CPAT had the same probability range which was "0 – 1". CPC used own probability ranges.

After that, we took the results from each table for each algorithm and created the different diagrams mentioned above to make the comparison between the algorithms.

6.2.1 ROC Curve Comparison for Test Set B

As previously mentioned, the Receiver Operating Characteristic curve, also known as ROC Curve, was the first diagram used for the comparison between the algorithms.

For Test Set B the ROC Curve results are shown in figure 6.10.

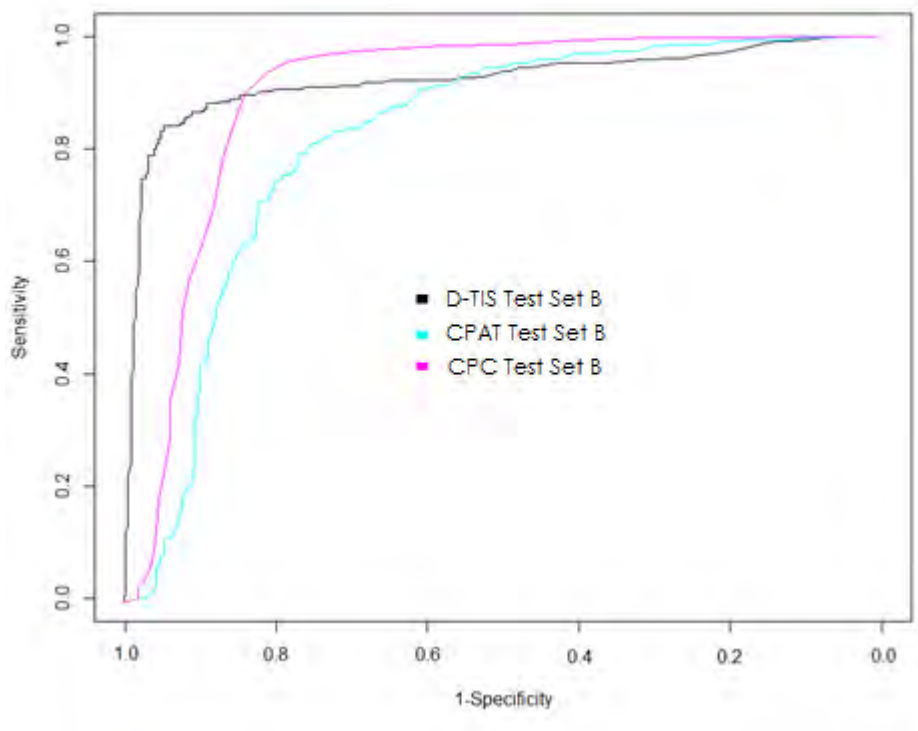


Figure 6.11: ROC Curve for Test Set B. This figure has been designed for the purpose of this thesis.

As shown in figure 6.10, D-TIS and CPC gave the highest AUC score. After these, CPAT followed. The area under the curve (AUC) for each program is given in the table below:

ALGORITHM	AUC
D-TIS	0.9236
CPAT	0.8204
CPC	0.9001

Table 6.7: The AUC scores from algorithms for Test Set B. This table has been designed for the purpose of this thesis.

To give more reliable results we needed to take into consideration the others diagrams also, especially "Precision-Recall".

6.2.2 Precision-Recall Curve for Test Set B

The "Precision – Recall" and "Recall – Precision" curves for the prediction of Test Set B, D-TIS returned higher results than the other programs. CPC and CPAT could predict sORFs genes but not so well. Their "Precision" after "0.7" threshold created a reverse curve. This paradox curve was created by a specific threshold and CPC and CPAT predicted a high ratio of False Positives (FP). For this reason, CPAT and CPC returned this paradox curve.

6.2.3 TP-TN Curve for the Test Set B

Another diagram used for the comparison of Test Set B between D-TIS, CPAT and CPC was the True Positives (TP) and True Negatives (TN) curve. The results of this diagram are described in figure 6.13.

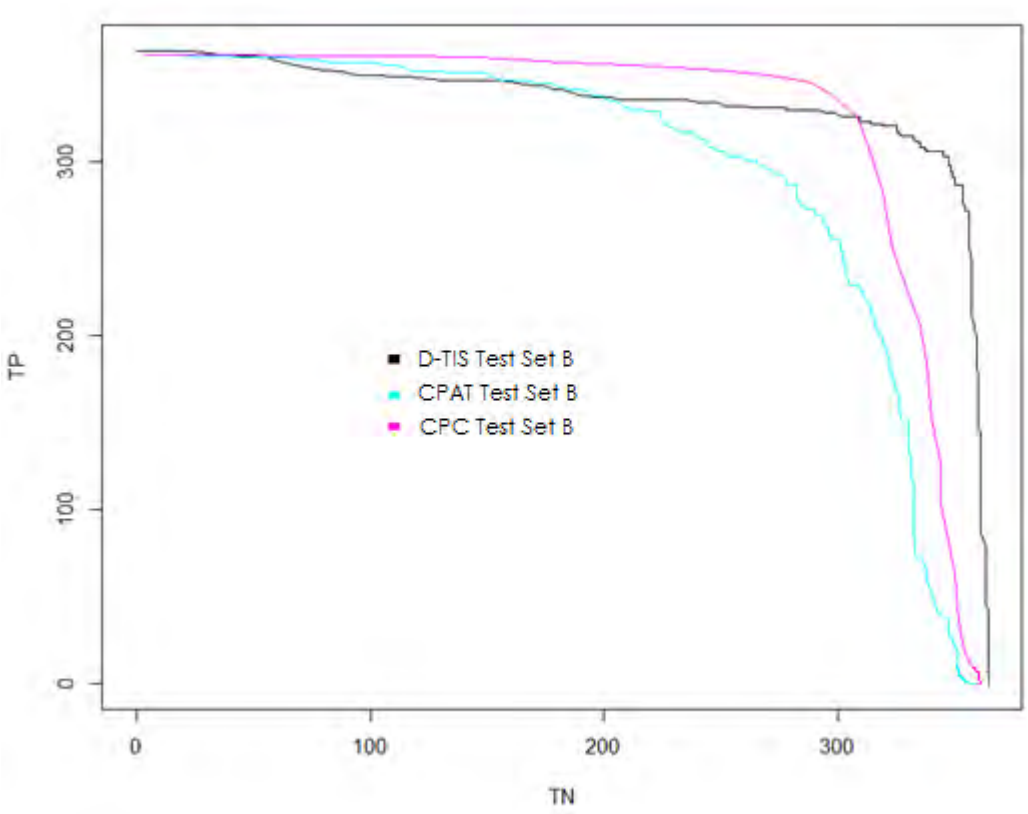


Figure 6.12: TP – TN curve for Test Set B. This figure has been designed for the purpose of this thesis.

As shown, CPC and D-TIS returned the highest scores. The next highest results returned from CPAT.

6.2.4 FP-FN Curve for Test Set B

A last diagram used to compare D-TIS versus CPAT and CPC for Test Set B. It was the False Positives (FP) and False Negatives (FN) curve. The diagram is presented in figure 5.24:

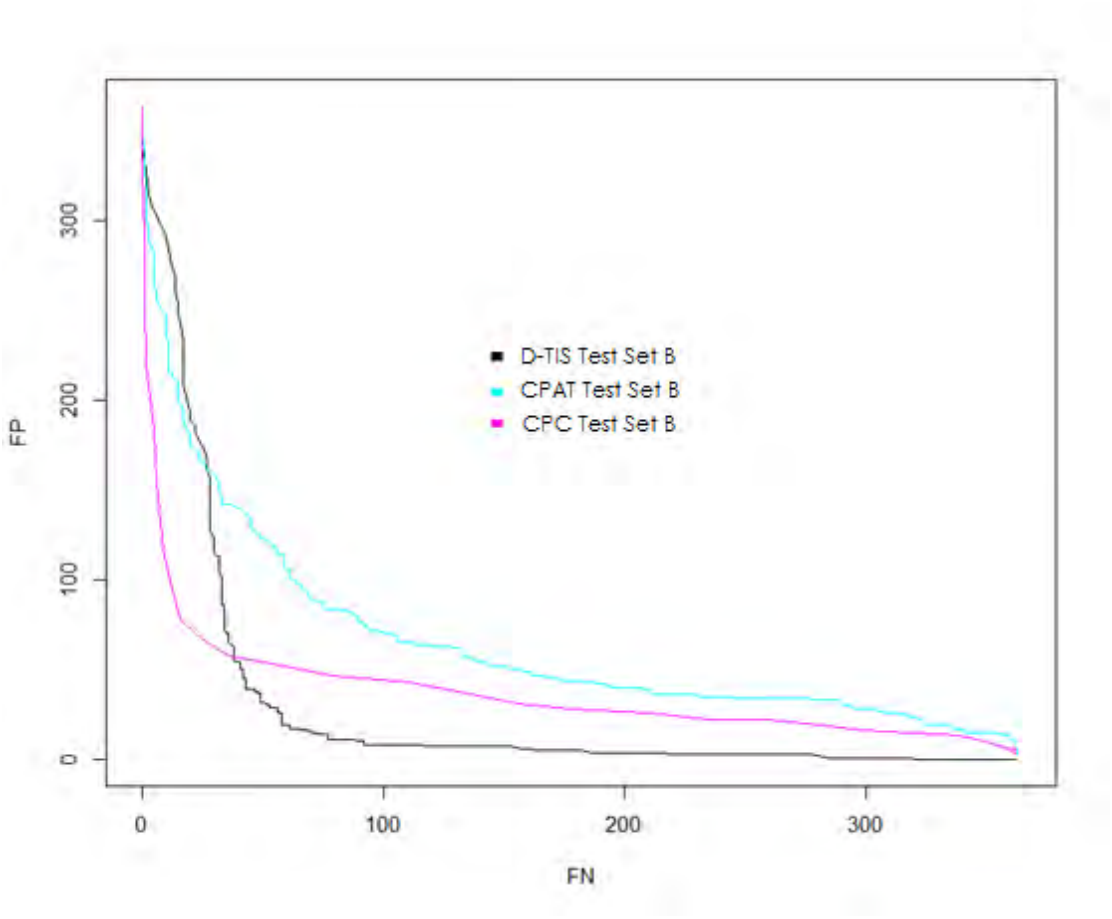


Figure 6.13: FP – FN for Test Set B. This figure has been designed for the purpose of this thesis.

As shown CPC and D-TIS returned the highest score. The next highest results returned from CPAT.

6.2.5 Thresholds Final Results for Test Set B

The diagrams showed that D-TIS returned the highest score for the prediction of Test Set B. The next highest results returned from CPC and last came CPAT.

To be more accurate, we must choose a specific threshold for the comparison of the algorithms for the prediction of the Test Set B or to present how each algorithm behaved with different thresholds. The tables below returned for a specific range of thresholds, the scores of the algorithms. As described before, D-TIS and CPAT had threshold range "0-1". CPC had different threshold ranges. CPC had range from "-1.37072" to "3.548819". More details are presented in the tables below:

D-TIS thresholds table for Test Set B:

Threshold	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
0.1	326	57	307	38	0.8511	0.8956	0.8434	0.8956	0.8695
0.2	315	34	330	49	0.9025	0.8653	0.9065	0.8653	0.8859
0.3	308	29	335	56	0.9139	0.8461	0.9203	0.8461	0.8832
0.4	306	23	341	58	0.9300	0.8406	0.9368	0.8406	0.8887
0.5	304	19	345	60	0.9411	0.8351	0.9478	0.8351	0.8914
0.6	298	17	347	66	0.9460	0.8186	0.9532	0.8186	0.8859
0.7	291	14	350	73	0.9540	0.7994	0.9615	0.7994	0.8804
0.8	283	11	353	81	0.9625	0.7774	0.9697	0.7774	0.8732
0.9	269	8	356	95	0.9711	0.7390	0.9780	0.7390	0.8585
1.0	2	0	364	362	1.0	0.0054	1.0	0.0054	0.5027

Table 6.8: Extend Rib & Cp thresholds table for Test Set B. This table has been designed for the purpose of this thesis

CPAT thresholds table for Test Set B:

Threshold	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
0.1	313	123	243	49	0.7178	0.8646	0.6639	0.8646	0.7637
0.2	272	76	290	90	0.7816	0.7513	0.7923	0.7513	0.7719
0.3	248	64	302	114	0.7948	0.6850	0.8251	0.6850	0.7554
0.4	215	52	314	147	0.8052	0.5939	0.8579	0.5939	0.7266
0.5	188	44	322	174	0.8103	0.5193	0.8797	0.5193	0.7005
0.6	157	40	326	205	0.7969	0.4337	0.8907	0.4337	0.6634
0.7	127	35	331	235	0.7839	0.3508	0.9043	0.3508	0.6291
0.8	88	34	332	274	0.7213	0.2430	0.9071	0.2430	0.5769
0.9	49	25	341	313	0.6621	0.1353	0.9316	0.1353	0.5357
1.0	0	5	361	362	0	0	0.9863	0	0.4958

Table 6.9: CPAT thresholds table for Test Set B. This table has been designed for the purpose of this thesis.

CPC thresholds table for Test set B:

Threshold	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
-1.37072	362	366	0	0	0.4972	1	0	1	0.4972
-1.026352	360	220	146	2	0.6206	0.9944	0.3989	0.9944	0.6950
-0.927961	356	154	212	6	0.6980	0.9834	0.5792	0.9834	0.7802
-0.829570	350	96	270	12	0.7847	0.9668	0.7377	0.9668	0.8516
-0.731179	339	69	297	23	0.8308	0.9364	0.8114	0.9364	0.8736
-0.583593	283	47	319	79	0.8575	0.7817	0.8715	0.7817	0.8269
-0.485202	228	37	329	134	0.8603	0.6298	0.8989	0.6298	0.7651
-0.386812	185	28	338	177	0.8685	0.5110	0.9234	0.5110	0.7184
-0.288421	127	22	344	235	0.8523	0.3508	0.9398	0.3508	0.6469
-0.091639	44	15	351	318	0.7457	0.1215	0.9590	0.1215	0.5425

Table 6.10: CPC thresholds table for Test Set B. This table has been designed for the purpose of this thesis.

A good method to finish with our comparison for Test Set B was to set as threshold the "0.5", which is the default threshold for machine learning.

In the table below we can see the comparison between algorithms for set threshold equal to "0.5". For the comparison of CPC used a threshold which had the best results for the program. CPC threshold was equal to "- 0.731179800000001".

Algorithm	*TP*	*FP*	*TN*	*FN*	*Precision*	*Recall*	*Specificity*	*Sensitivity*	*Acc*
D-TIS	304	19	345	60	**0.9411**	0.8351	**0.9478**	0.8351	**0.8914**
CPAT	188	44	322	174	0.8103	0.5193	0.8797	0.5193	0.7005
CPC	339	69	297	23	0.8308	0.9364	0.8114	**0.9364**	0.8736

Table 6.11: Algorithms final results for Test Set B. This table has been designed for the purpose of this thesis.

The table 6.11 shows that D-TIS returned the highest "Accuracy". The other programs, except for CPC, gave low results. The results are discussed further in the next chapter.

Chapter 7
Discussion and Conclusion

7.1 Discussion

As shown from different kinds of diagrams and from the thresholds tables, D-TIS returned the highest scores for the prediction of the Test Set A and Test Set B.

A big challenge for the TIS tools are the discrimination of sORFs genes from ncRNAs. sORFs, as described, were small Open Reading Frames (ORF) which produce proteins and their length was smaller than 300 nucleotides. It is very difficult to distinguish these genes from non-coding sequences. The reason is that non-coding sequences have in average the same length as sORFs. sORFs because of their distinctiveness with their length and because their existence was detected recently. There are yet no reliable tools which can distinguish sORFs from non-coding sequences with high accuracy.

The most significant challenge for D-TIS was not only to distinguish with high accuracy the Test Set A but to distinguish with high accuracy the Test Set B which consists from 364 sORFs as positive set and 364 non-coding sequences upstream of cDNA region.

7.1.1 Discussion about the Test Set A prediction

D-TIS for the prediction of Test Set A, along with CPAT, returned the highest scores. D-TIS had the highest "Accuracy" of all the other methods, equal to "0.9367". Furthermore, D-TIS had high "Sensitivity" score equal to "0.9495" and "Specificity" equal to "0.9239" for threshold = "0.5". D-TIS returned the second highest score for "Precision" = 0.9258. Also, D-TIS could distinguish with high ratio the "True Positives" from "True Negatives". The disadvantage was that D-TIS produced a significant rate of "False Positives". That means that it predicted as "Positives" a rate of "Negatives".

CPAT for threshold equal to "0.5" returned the highest "Sensitivity" score, equal to "0.9757". CPAT could predict with high ratio the "True Positives" from "Negatives". At the same time, it predicted with high rate

the “False Positives”. That means that CPAT predicted “True Negatives” as “Positives”. That is the reason why it had the lowest “Specificity” score, equal to “0.8562” and one of the lowest “Accuracy” scores equal to “0.9159”, compared to the other methods. CPAT predicted the positive set with extreme positive probability score equal to “1”. For that reason, there was a high score of AUC, equal to “0.9747”.

CPC had a different range for threshold. But we tried to find an approximate threshold to make the comparison between the methods. CPC as shown from the “ROC” curve gave the third highest “AUC” score equal to “0.9738”. CPC for a specific threshold = “- 0.224236” gave the highest “Specificity” score = “0.9418”. That means that CPC could predict “True Negatives” with higher accuracy than “True Positives” but produced a high rate of “False Negatives” and that created low “Sensitivity” = “0.9092”. Furthermore, CPC had the higher “Precision” score of all methods equal to “0.9398”.

PLEK, as CPC had different thresholds range. But we tried, approximately, to find a threshold to make the comparison between the methods. That threshold was equal to “0.28633”. PLEK returned the lowest “Accuracy” score = “0.8987” compared to the other methods. Also, it had the lowest rate of “True Positives”. CPC could not predict with high accuracy the “True Positives” and produced a high rate of “False Negatives” and “False Positives”. PLEK had the lowest “Sensitivity” score = “0.8869” and the second lowest “Specificity” score equal to “0.9105”.

The final conclusion about Test Set A is that we created a new method, called D-TIS to distinguish coding sequences from non-coding sequences. The D-TIS method compared to the state-of-the-art methods for coding potential, such as CPAT succeeded in reaching and surpassing the rates of success even of the best algorithms. Also, D-TIS keeps a balance between “Specificity” and “Sensitivity” and for this reason it returned the highest “Accuracy” equal to “0.9367”.

7.1.2 Discussion about the Test Set B prediction

As mentioned above, a new bigger challenge is to distinguish sORFs sequences from non-coding sequences.

For the prediction of the Test Set B, D-TIS returned the highest scores. More specifically, for threshold = "0.5" D-TIS gave the highest "Accuracy" score equal to "0.8914", the highest "Precision" and "Specificity" scores, respectively equal to "0.9411" and "0.9478". That means that D-TIS could distinguish with high accuracy the "True Negatives" from "True Positives". The disadvantage was that D-TIS produced a significant rate of "False Negatives". Furthermore, D-TIS returned the highest "AUC" score for the "ROC" curve equal to "0.9236".

CPAT, as described in the diagrams and tables above, could not distinguish the Test Set B for different thresholds with high rates. CPAT, and more specifically for the default machine learning threshold which is equal to "0.5", had extremely low rates. There was extremely low "Sensitivity" = "0.5193" and low "Accuracy" = "0.7005". CPAT in contrast with "Sensitivity" has high "Specificity" = "0.8797". CPAT "AUC" was "0.8204", which was the last rate score in comparison with the methods. From "Precision-Recall" curve it turns out that CPAT had a problem with the prediction of Test Set B. More specifically, CPAT from threshold equal to "0.7" predicted with high rate "False Negatives" and for this reason it produced the paradox with the "Precision" curve. For threshold "0.7" the predicted "True Positives" were equal to "127" and the "False Negatives" equal to "235" genes. The conclusion about CPAT is that CPAT was not trustworthy for the prediction of Test Set B.

CPC returned the second highest score for the prediction of Test Set B for an approximate threshold. More specifically, CPC produced the highest "Sensitivity" = "0.9364" score, it had low "Specificity" score equal to "0.8114" and "Accuracy" score = "0.8736". From the analysis of the results it turns out that CPC could predict with high rate the "True Positives" from "True Negatives". But in contrast with D-TIS, CPC produced a lot of "False Positives" rates. From the "Precision-Recall" curve it turns out that CPC had a problem with the prediction of Test Set B. More specifically, CPC, from threshold "-0.485202" and then, as did CPAT, predicted with high rate "False Negatives". For this reason, it produced the paradox with the "Precision" curve. Moreover, for threshold "-0.485202" the predicted "True Positives" were equal to "228" genes and "False Negatives" equal to "134" genes.

The final conclusion about the Test Set B is that we created a new method, called D-TIS to distinguish coding sequences from non-coding sequences. That new method was compared with the state-of-the-art

methods for coding potential, such as CPAT. It succeeded in reaching and surpassing the rates of success of the best algorithms. And for this reason, we used D-TIS to distinguish small Open Reading Frame sequences from non-coding sequences. The results showed that D-TIS which was trained to distinguish coding sequences from cDNA from non-coding sequences could distinguish equally well and sORFs from non-coding sequences. The other methods like CPAT and CPC could not distinguish sORF sequence from non-coding sequences so well as D-TIS.

7.2 Conclusion

The final conclusion of our thesis is that we introduced an integrated computational pipeline based on a machine learning technique and more specifically in a support vector machine on GPU-LIBSVM method. The pipeline was designed to distinguish coding DNA sequences from non-coding sequences. For the method we created an algorithm, called D-TIS which is based on two biologically meaningful sequence features.

For the test of D-TIS used two test sets. The first test set consists of 1.565 positive and 1565 negative sequences out of 11.565. This test set, called "Test Set A".

The second test set built from 354 sORFs which derived from the supplementary material of (Mackowiak, Zauber et al. 2015) and 354 non-coding sequences from the upstream flank of cDNA. This set, called "Test Set B".

The results showed that D-TIS could distinguish the two sets with high rates. That makes D-TIS competitive compared to other programs which are used for coding potential prediction.

Our thesis is a new integrated computational pipeline which consists of exclusively two biologically meaningful sequence features based on GPU-LIBSVM. The method is sufficiently competitive compared to the other coding potential programs which are used to distinguish coding from non-coding sequences, like CPAT, CPC and PLEK. The most significant point is that our pipeline can be used to distinguish equally well coding sequences from non-coding sequences and small Open Reading Frame sequences from non-coding. A last detail is that our pipeline can be used to distinguish with high success rates and small Open Reading Frame sequences from non-coding without being trained to distinguish them.

References

Ahlquist, P. (2002). "RNA-dependent RNA polymerases, viruses, and RNA silencing." Science **296**(5571): 1270-1273.
Alberts, B. (2008). Molecular biology of the cell. New York, Garland Science.
Amaral, P. P. and J. S. Mattick (2008). "Noncoding RNA in development." Mamm Genome **19**(7-8): 454-492.
Arrial, R. T., R. C. Togawa and M. Brigido Mde (2009). "Screening non-coding RNAs in transcriptomes from neglected species using PORTRAIT: case study of the pathogenic fungus Paracoccidioides brasiliensis." BMC Bioinformatics **10**: 239.
Badger, J. H. and G. J. Olsen (1999). "CRITICA: coding region identification tool invoking comparative analysis." Mol Biol Evol **16**(4): 512-524.
Barciszewski, J., B. F. C. Clark and SpringerLink (Online service) (1999). RNA Biochemistry and Biotechnology. NATO Science Series, Series 3: High Technology,. Dordrecht, Springer Netherlands : Imprint: Springer,**:** 1 online resource.
Batey, R. T. (2006). "Structures of regulatory elements in mRNAs." Curr Opin Struct Biol **16**(3): 299-306.
Berg, J. M., J. L. Tymoczko and L. Stryer (2007). Biochemistry. New York, W. H. Freeman.
Blevins, T., R. Rajeswaran, P. V. Shivaprasad, D. Beknazariants, A. Si-Ammour, H. S. Park, F. Vazquez, D. Robertson, F. Meins, Jr., T. Hohn and M. M. Pooggin (2006). "Four plant Dicers mediate viral small RNA biogenesis and DNA virus induced silencing." Nucleic Acids Res **34**(21): 6233-6246.
Campbell, N. A. and J. B. Reece (2005). Biology. San Francisco, Pearson, Benjamin Cummings.
Cantara, W. A., P. F. Crain, J. Rozenski, J. A. McCloskey, K. A. Harris, X. Zhang, F. A. Vendeix, D. Fabris and P. F. Agris (2011). "The RNA Modification Database, RNAMDB: 2011 update." Nucleic Acids Res **39**(Database issue): D195-201.
Cavaille, J., M. Nicoloso and J. P. Bachellerie (1996). "Targeted ribose methylation of RNA in vivo directed by tailored antisense RNA guides." Nature **383**(6602): 732-735.
Cooper, G. M. and R. E. Hausman (2004). The cell : a molecular approach. Washington, D.C.

Sunderland, Mass., ASM Press ;

Sinauer Associates.
Correia, J. J. and W. F. Stafford (2015). "Sedimentation Velocity: A Classical Perspective." Methods Enzymol **562**: 49-80.
Crick, F. (1970). "Central dogma of molecular biology." Nature **227**(5258): 561-563.
Daros, J. A., S. F. Elena and R. Flores (2006). "Viroids: an Ariadne's thread into the RNA labyrinth." EMBO Rep **7**(6): 593-598.
Deonier, R. C., S. Tavaré, M. S. Waterman and SpringerLink (Online service) (2005). Computational Genome Analysis An Introduction. New York, NY, Springer Science+Business Media, Inc.,.
Elliott, M. S. and R. W. Trewyn (1984). "Inosine biosynthesis in transfer RNA by an enzymatic insertion of hypoxanthine." J Biol Chem **259**(4): 2407-2410.
Girard, A., R. Sachidanandam, G. J. Hannon and M. A. Carmell (2006). "A germline-specific class of small RNAs binds mammalian Piwi proteins." Nature **442**(7099): 199-202.
Griffiths, A. J. F. (2008). Introduction to genetic analysis. New York, W.H. Freeman and Co.
Gueneau de Novoa, P. and K. P. Williams (2004). "The tmRNA website: reductive evolution of tmRNA in plastids and other endosymbionts." Nucleic Acids Res **32**(Database issue): D104-108.
Guo, H., N. T. Ingolia, J. S. Weissman and D. P. Bartel (2010). "Mammalian microRNAs predominantly act to decrease target mRNA levels." Nature **466**(7308): 835-840.

Hansen, J. L., A. M. Long and S. C. Schultz (1997). "Structure of the RNA-dependent RNA polymerase of poliovirus." Structure **5**(8): 1109-1122.
Hatzigeorgiou, A. G. (2002). "Translation initiation start prediction in human cDNAs with high accuracy." Bioinformatics **18**(2): 343-350.
Heard, E., F. Mongelard, D. Arnaud, C. Chureau, C. Vourc'h and P. Avner (1999). "Human XIST yeast artificial chromosome transgenes show partial X inactivation center function in mouse embryonic stem cells." Proc Natl Acad Sci U S A **96**(12): 6841-6846.
Hermann, T. and D. J. Patel (2000). "RNA bulges as architectural and recognition motifs." Structure **8**(3): R47-54.
Horvath, P. and R. Barrangou (2010). "CRISPR/Cas, the immune system of bacteria and archaea." Science **327**(5962): 167-170.
Horwich, M. D., C. Li, C. Matranga, V. Vagin, G. Farley, P. Wang and P. D. Zamore (2007). "The Drosophila RNA methyltransferase, DmHen1, modifies germline piRNAs and single-stranded siRNAs in RISC." Curr Biol **17**(14): 1265-1272.
Johnson, Z. I. and S. W. Chisholm (2004). "Properties of overlapping genes are conserved across microbial genomes." Genome Res **14**(11): 2268-2272.
Kalendar, R., C. M. Vicient, O. Peleg, K. Anamthawat-Jonsson, A. Bolshoy and A. H. Schulman (2004). "Large retrotransposon derivatives: abundant, conserved but nonautonomous retroelements of barley and related genomes." Genetics **166**(3): 1437-1450.
Kawano, Y., S. Neeley, K. Adachi and H. Nakai (2013). "An experimental and computational evolution-based method to study a mode of co-evolution of overlapping open reading frames in the AAV2 viral genome." PLoS One **8**(6): e66211.
Khoury, G. A., R. C. Baliban and C. A. Floudas (2011). "Proteome-wide post-translational modification statistics: frequency analysis and curation of the swiss-prot database." Sci Rep **1**.
King, T. H., B. Liu, R. R. McCully and M. J. Fournier (2003). "Ribosome structure and activity are altered in cells lacking snoRNPs that form pseudouridines in the peptidyl transferase center." Mol Cell **11**(2): 425-435.
Kiss, T. (2001). "Small nucleolar RNA-guided post-transcriptional modification of cellular RNAs." EMBO J **20**(14): 3617-3622.
Kong, L., Y. Zhang, Z. Q. Ye, X. Q. Liu, S. Q. Zhao, L. Wei and G. Gao (2007). "CPC: assess the protein-coding potential of transcripts using sequence features and support vector machine." Nucleic Acids Res **35**(Web Server issue): W345-349.
Labrou, N. E. (2014). "Protein purification: an overview." Methods Mol Biol **1129**: 3-10.
Lee, J. C. and R. R. Gutell (2004). "Diversity of base-pair conformations and their occurrence in rRNA structure and RNA structural motifs." J Mol Biol **344**(5): 1225-1249.
Li, A., J. Zhang and Z. Zhou (2014). "PLEK: a tool for predicting long non-coding RNAs and messenger RNAs based on an improved k-mer scheme." BMC Bioinformatics **15**: 311.
Mackowiak, S. D., H. Zauber, C. Bielow, D. Thiel, K. Kutz, L. Calviello, G. Mastrobuoni, N. Rajewsky, S. Kempa, M. Selbach and B. Obermayer (2015). "Extensive identification and analysis of conserved small ORFs in animals." Genome Biol **16**: 179.
Madigan, M. T., M. T. Madigan and T. D. Brock (2009). Brock biology of microorganisms. San Francisco, CA, Pearson/Benjamin Cummings.
Mathews, D. H., M. D. Disney, J. L. Childs, S. J. Schroeder, M. Zuker and D. H. Turner (2004). "Incorporating chemical modification constraints into a dynamic programming algorithm for prediction of RNA secondary structure." Proc Natl Acad Sci U S A **101**(19): 7287-7292.
Mattick, J. S. (2001). "Non-coding RNAs: the architects of eukaryotic complexity." EMBO Rep **2**(11): 986-991.
Mattick, J. S. (2003). "Challenging the dogma: the hidden layer of non-protein-coding RNAs in complex organisms." Bioessays **25**(10): 930-939.

McCulloch, S. D. and T. A. Kunkel (2008). "The fidelity of DNA synthesis by eukaryotic replicative and translesion synthesis polymerases." Cell Res **18**(1): 148-161.
Mikkola, S., E. Stenman, K. Nurmi, E. Yousefi-Salakdeh, R. Stromberg and H. Lonnberg (1999). "The mechanism of the metal ion promoted cleavage of RNA phosphodiester bonds involves a general acid catalysis by the metal aquo ion on the departure of the leaving group." Journal of the Chemical Society, Perkin Transactions 2(8): 1619-1626.
Min, X. J., G. Butler, R. Storms and A. Tsang (2005). "OrfPredictor: predicting protein-coding regions in EST-derived sequences." Nucleic Acids Res **33**(Web Server issue): W677-680.
Moghal, A., K. Mohler and M. Ibba (2014). "Mistranslation of the genetic code." FEBS Lett **588**(23): 4305-4310.
Nissen, P., J. Hansen, N. Ban, P. B. Moore and T. A. Steitz (2000). "The structural basis of ribosome activity in peptide bond synthesis." Science **289**(5481): 920-930.
Nudler, E. and M. E. Gottesman (2002). "Transcription termination and anti-termination in E. coli." Genes Cells **7**(8): 755-768.
Podlevsky, J. D., C. J. Bley, R. V. Omana, X. Qi and J. J. Chen (2008). "The telomerase database." Nucleic Acids Res **36**(Database issue): D339-343.
Pushparaj, P. N., J. J. Aarthi, S. D. Kumar and J. Manikandan (2008). "RNAi and RNAa--the yin and yang of RNAome." Bioinformation **2**(6): 235-237.
Robertson, M. P. and G. F. Joyce (2012). "The origins of the RNA world." Cold Spring Harb Perspect Biol **4**(5).
Ross, J. F. and M. Orlowski (1982). "Growth-rate-dependent adjustment of ribosome function in chemostat-grown cells of the fungus Mucor racemosus." J Bacteriol **149**(2): 650-653.
Rossi, J. J. (2004). "Ribozyme diagnostics comes of age." Chem Biol **11**(7): 894-895.
Rother, M. and J. A. Krzycki (2010). "Selenocysteine, pyrrolysine, and the unique energy metabolism of methanogenic archaea." Archaea **2010**.
Russell, P. J. (2010). iGenetics : a molecular approach. San Francisco, Benjamin Cummings.
Scotto, L. and R. K. Assoian (1993). "A GC-rich domain with bifunctional effects on mRNA and protein levels: implications for control of transforming growth factor beta 1 expression." Mol Cell Biol **13**(6): 3588-3597.
Sedova, A. and N. K. Banavali (2015). "RNA approaches the B-form in stacked single strand dinucleotide contexts." Biopolymers.
Sontheimer, E. J. and R. W. Carthew (2005). "Silence from within: endogenous siRNAs and miRNAs." Cell **122**(1): 9-12.
St Laurent, G., D. Shtokalo, M. R. Tackett, Z. Yang, T. Eremina, C. Wahlestedt, S. Urcuqui-Inchima, B. Seilheimer, T. A. McCaffrey and P. Kapranov (2012). "Intronic RNAs constitute the major fraction of the non-coding RNA in mammalian cells." BMC Genomics **13**: 504.
Steitz, T. A. and J. A. Steitz (1993). "A general two-metal-ion mechanism for catalytic RNA." Proc Natl Acad Sci U S A **90**(14): 6498-6502.
Toyama, B. H. and M. W. Hetzer (2013). "Protein homeostasis: live long, won't prosper." Nat Rev Mol Cell Biol **14**(1): 55-61.
Vazquez, F., H. Vaucheret, R. Rajagopalan, C. Lepers, V. Gasciolli, A. C. Mallory, J. L. Hilbert, D. P. Bartel and P. Crete (2004). "Endogenous trans-acting siRNAs regulate the accumulation of Arabidopsis mRNAs." Mol Cell **16**(1): 69-79.
Wagner, E. G., S. Altuvia and P. Romby (2002). "Antisense RNAs in bacteria and their genetic elements." Adv Genet **46**: 361-398.
Wang, L., H. J. Park, S. Dasari, S. Wang, J. P. Kocher and W. Li (2013). "CPAT: Coding-Potential Assessment Tool using an alignment-free logistic regression model." Nucleic Acids Res **41**(6): e74.
Watson, J. D. (2007). Recombinant DNA : genes and genomes - a short course. New York, N.Y., W.H. Freeman and Company.

Watson, J. D. and F. H. Crick (1953). "Molecular structure of nucleic acids; a structure for deoxyribose nucleic acid." Nature **171**(4356): 737-738.
Whitehead, K. A., J. E. Dahlman, R. S. Langer and D. G. Anderson (2011). "Silencing or stimulation? siRNA delivery and the immune system." Annu Rev Chem Biomol Eng **2**: 77-96.
Xie, J., M. Zhang, T. Zhou, X. Hua, L. Tang and W. Wu (2007). "Sno/scaRNAbase: a curated database for small nucleolar RNAs and cajal body-specific RNAs." Nucleic Acids Res **35**(Database issue): D183-187.
Yates, A., W. Akanni, M. R. Amode, D. Barrell, K. Billis, D. Carvalho-Silva, C. Cummins, P. Clapham, S. Fitzgerald, L. Gil, C. G. Giron, L. Gordon, T. Hourlier, S. E. Hunt, S. H. Janacek, N. Johnson, T. Juettemann, S. Keenan, I. Lavidas, F. J. Martin, T. Maurel, W. McLaren, D. N. Murphy, R. Nag, M. Nuhn, A. Parker, M. Patricio, M. Pignatelli, M. Rahtz, H. S. Riat, D. Sheppard, K. Taylor, A. Thormann, A. Vullo, S. P. Wilder, A. Zadissa, E. Birney, J. Harrow, M. Muffato, E. Perry, M. Ruffier, G. Spudich, S. J. Trevanion, F. Cunningham, B. L. Aken, D. R. Zerbino and P. Flicek (2016). "Ensembl 2016." Nucleic Acids Res **44**(D1): D710-716.
Yu, Q. and C. D. Morrow (2001). "Identification of critical elements in the tRNA acceptor stem and T(Psi)C loop necessary for human immunodeficiency virus type 1 infectivity." J Virol **75**(10): 4902-4906.

Appendices

Appendix A How to install CPAT

What does CPAT need to work?

CPAT can work from its web interface or the users can download it and install it on a personal computer. The users can download from this link http://rna-cpat.sourceforge.net/#install-cpat-to-local-computer the last version of CPAT only for the operating system UNIX or can use the online edition from the link below http://lilab.research.bcm.edu/cpat/.

CPAT needs some other programs to be installed to work properly. The users need to download and install the following programs:

1. Python 2.7 ++(In this thesis we used Python 2.7.9)
2. NumPy 1.9.2
3. Cython 0.22
4. Nose 1.3.6
5. R-studio
6. CPAT

CPAT installation

After seeing what CPAT needs, let's see how the users can install the above programs. First, the users need to install the "Python" and after that all other programs. Let's see the installation step by step.

How to install Python for CPAT

Let's install "Python". First, the users need to open the "Linux" terminal and follow the steps below:

1. Download Python-2.7.9
2. https://www.python.org/downloads
3. Unzip the downloaded file to a specific location
4. Go to the location with the command
5. cd /the users location/Python-2.7.9

6. Install Python with the command
7. sudo ./configure
8. sudo make
9. sudo make install

After these steps, the installation of "Python" has finished. Attention with the command "sudo", you need to insert your password.

How to install NumPy for CPAT

Since the users have finished the installation of "Python", they need to install the "NumPy". The users need to open the terminal and follow the steps:

1. Download NumPy 1.9.2 from the link bellow
   - ://sourceforge.net/projects/numpy/files/NumPy/1.9.2/numpy-1.9.2.tar.gz/download
2. Unzip the downloaded file to a specific location
3. Go to the specific location and find the unzip folder
4. Cd/users specific location/NumPy 1.9.2
5. sudo python setup.py install

How to install Cython for CPAT

The users need to install the "Cython 0.22" version for the installation of CPAT. Let's see the process step by step, but first open "Linux" terminal:

1. Download Cython from the following link
   - http://cython.org/release/Cython-0.22.tar.gz
2. Unzip the downloaded file to a specific location
3. Go to the specific location and find the unzip folder
4. Cd/users specific location/Cython 0.22
5. sudo python setup.py install.

How to install Nose for CPAT

1. Download Nose 1.3.6 from the following link
   - https://pypi.python.org/packages/source/n/nose/nose-1.3.6.tar.gz#md5=0ca546d81ca8309080fc80cb389e7a16
2. Unzip the downloaded file to a specific location
3. Go to the specific location and find the unzip folder

4. cd/users specific location/Nose 1.3.6
5. sudo python setup.py install

How to install R-studio for CPAT

1. Download R-studio from the link bellow
   - http://cran.rstudio.com
2. Download R for Linux
3. Choose your operating system (etc. Ubuntu)
4. Choose mirror for download
5. deb http://<my.favorite.cran.mirror>/bin/linux/ubuntu vivid/
6. Open terminal and write the following commands
7. sudo apt-get update
8. sudo apt-get install r-base
9. sudo apt-get install r-base-dev

How to install CPAT for CPAT

When the users have finished with the installation of the above programs, they are ready to install CPAT. First, they need to open Linux terminal and proceed as follows:

1. Download CPAT-1.2.2 from the link below
   - http://rna-cpat.sourceforge.net/#install-cpat-to-local-computer
2. Unzip the downloaded file
3. tar zxf CPAT-1.2.2.tar.gz
4. Go to the folder
5. cd CPAT-1.2.2

Install the CPAT.

1. sudo python setup.py install
2. cd CPAT-1.2.2/test
3. On folder test the user insert the FASTA and BED files with the following format:
   - For FASTA files:
     - Name.fa
   - For BED files:
     - Name.bed
4. Execute the program. User needs to stay to the folder test and run the following commands:

- cpat.py -g Name.fa -d ../dat/Human_logitModel.RData -x ../dat/Human_Hexamer.tsv -o output2
- head output2

With the two last commands the users can give the FASTA file which is needed for the test and can change the output2 into the file name by which they want to print the results. CPAT returns three files: output2, output2.dat and output2.r.

Appendix B How to install CPC

What does CPC need to work?

CPC can work from its web interface or users can download it and install it on their personal computer. Users can download the program from this link: http://cpc.cbi.pku.edu.cn/download The latest version of CPC only works for the operating system "UNIX". In addition, the users can use the online version of the program from the link below http://cpc.cbi.pku.edu.cn/.

CPC also needs NCBI BLAST package to work. So the users need to download and install the NCBI BLAST database.

CPC installation

After seeing what CPC needs, let's see how the users can install the above database. Let's take the installation step by step.

How to install NCBI BLAST package for CPC

Let's install NCBI BLAST package. First, the users need to open the Linux terminal and follow these steps:

1. Download NCBI BLAST package with a browser:
    - ftp://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/
2. Choose version:
    - ncbi-blast-2.2.30+-x64-linux.tar.gz
3. Go on Linux terminal and find the downloaded file and unzip and follow the bellow commands:
    - tar zxvpf ncbi-blast-2.2.30+-x64-linux.tar.gz
    - export PATH="$PATH:$HOME/ncbi-blast-2.2.30+/bin"
    - mkdir ./ncbi-blast-2.2.30+/db
    - cd ncbi-blast-2.2.30+/db
    - ftp ftp.ncbi.nlm.nih.gov
4. Name (ftp.ncbi.nlm.nih.gov:user): anonymous
5. Asks for name, choose anonymous
6. Password:konstantinosli*******@gmail.com
    - Asks for password, put a valid E-mail
7. cd blast/db
8. bin

9. get refseq_protein.00.tar.gz
10. bye
11. tar zxvpf refseq_protein.00.tar.gz
12. ls -ltr refseq_protein.00*
13. rm refseq_protein.00.tar.gz

Once the users have finished with the commands above, they must go and copy the NCBI BLAST package into the CPC folder.

How to install CPC for CPC

Now the users must install the CPC only for "Linux" operating system. Let's install CPC. First, the users need to download CPC. After that, install it from the terminal. Let's see the instructions step by step:

1. Download CPC from the following link:
    - http://cpc.cbi.pku.edu.cn/download
2. Find the downloaded file
3. Open terminal and write the following commands:
    - gzip -dc cpc-0.9.tar.gz | tar xf –
    - cd cpc-0.9
    - export CPC_HOME="$PWD"
    - cd libs/libsvm
    - gzip -dc libsvm-2.81.tar.gz | tar xf –
    - cd libsvm-2.81
    - make clean && make
    - cd ../..
    - gzip -dc estate.tar.gz | tar xf –
    - cd estate
    - make clean && make
4. Go to the base NCI BLAST database package ncbi-blast-2.2.30+db and copy the folder db into the folder cpc/data and rename it into prot_db
5. Go to terminal and continue with the commands bellow:
    - cd $CPC_HOME/data
    - formatdb -i (your_fasta_file) -p T -n prot_db
    - The user must put FASTA file into cpc/data folder and with these style: "Name.fasta"
    - cd $CPC_HOME
    - bin/run_predict.sh input_seq result_in_table working_dir result_evidence

"input_seq" stands for the FASTA which needs prediction, "result_in_table" stands for the file where the results will print out,

"working dir" stands for the folder "ncbi-blast-2.2.30+" and the "result_evidence" is the file with the result evidence as printed out. CPC will return three files to the user: result_evidence.homo, result_evidence.orf and result_in_table.

Appendix C How to install PLEK

What does PLEK need to work?

PLEK is easy to install. It is an open-source software for "Linux". Let's see how to install PLEK.

How to install PLEK

As mentioned above, PLEK works only for "Linux". So, let's see the steps which are needed to install the software.

1. Download the software from this link:
    - http://sourceforge.net/projects/plek/files/latest/downld
2. Unzip the downloaded file
3. Open Linux terminal and find the destination of the unzip file and follow the command:
    - python PLEK_setup.py

1. Put FASTA files with these form "Name.fa" and put it into the PLEK folder
2. Run PLEK with the following command:
    - python PLEK.py -fasta FastaName.fa -out predicted -thread 10

    Or

    - python PLEK.py -fasta FastaName.fa -out predicted -thread 10 -minlength 150

    PLEK returns only one text file with the default name "predicted".

Appendix D How to install PORTRAIT

What does PORTRAIT need to work?

PORTRAIT needs some extra programs to work. First the users need to download and install PORTRAIT. Afterwards, users need to install three other extra programs, "LIBSVM 2.84", "CAST 1.0" and "ANGLE". PORTRAIT is a software which works for operating system of "Linux". Let's see how to install PORTRAIT software on "Linux".

How to install PORTRAIT

PORTRAIT can be installed only for "Linux". So the users need to download PORTRAIT software and the three extra programs to be in position to work with them. Let's see how to install the programs.

How to install LIBSVM 2.84 for PORTRAIT

1. Download the LIBSVM 2.84 software from the link above.
2. Find the downloaded file.
3. Open Linux terminal and unzip the file with this command:
    - gunzip -c libsvm-2.84.tar.gz | tar xvf -
4. Wait to download the PORTRAIT software and then copy the file and put it into the PORTRAIT folder.
5. Afterwards install the LIBSVM with the following commands:
    - cd libsvm-2.84
    - make

How to install CAST 1.0 for PORTRAIT

1. Download the CAST 1.0 software from the link:
    - http://bioinformatics.cenargen.embrapa.br/portrait/download/
2. Find the downloaded file and unzip it.
3. Wait to download the PORTRAIT software and then copy the file and put it into the PORTRAIT folder.

How to install ANGLE for PORTRAIT

1. Download the CAST 1.0 software from the link:
    - http://bioinformatics.cenargen.embrapa.br/portrait/download/
2. Find the downloaded file and unzip it.
3. Wait to download the PORTRAIT software and after that copy the file and put it into the PORTRAIT folder.

How to install PORTRAIT for PORTRAIT

1. Download PORTRAIT from the above link:
2. http://bioinformatics.cenargen.embrapa.br/portrait/download/
3. Open Linux terminal find the file and unzip it with this command:
4. gunzip -c portrait-1.1.tar.gz | tar xvf –
5. Install the program with the following command:
6. perl portrait-1.1.pl
7. give the PORTRAIT folder direction
8. give the LIBVSM-2.84 folder direction
9. give the CAST 1.0 folder direction
10. give the ANGLE folder direction
11. Run again the following command
12. perl portrait-1.1.pl
13. Run the program for prediction with this command:
14. perl portrait-1.1.pl -i FastaName.fasta -s(or -c or -a)

The users need to save the FASTA file with a specific format like "Name.fasta" inside the PORTRAIT folder. PORTRAIT software has finished with the prediction and returns three to sixteen text files to users, depending on the type of genes.

Appendix E How to install OrfPredictor

What does OrfPredictor need to work?

OrfPredictor can easily be installed on a personal computer. The only thing needed is the OrfPredictor program.

How to install OrfPredictor

The first thing to do to install OrfPredictor is to ask the authors of OrfPredictor to send the software for installation. Next, the users must follow the steps below:

1. On Linux find the downloaded file
2. Unzip the file
3. Open Linux terminal and find with cd command the destination of the unzip file
4. Install the program with the following commands:
5. chmod 777 ./extractCDS.pl
6. chmod 777 ./OrfPredictor_web3.pl
7. Run the program for prediction with the following command:
8. perl ./OrfPredictor_web3.pl FastaName.fasta BLASTX bFlog strand Email Evalue output url

The users need to insert the FASTA file into the OrfPredictor folder with the format "Name.fasta". OrfPredictor returns three text files to users, "noORF_id.txt", "ORF6frame.txt" and "output.txt". But before finishing with the OrfPredictor, let's see the last command. In the last command the users need to choose some info. This info is:

1. BLASTX
    - If we have a folder with BLASTX then we must give the name of the folder

      OR

    - Give a void file as BLASTX
2. bFlog
    - If we have BLASTX give for the bFlog the price =1

OR

- bFlog =0

3. strand
    - Search for both with the command both
    - Search for + with the command +
    - Search for – with the command –
4. Email
    - Give a valid E-mail
5. Evalue
    - Keep it as is
6. Output
    - Keep it as is
7. Url
    - Write www.yoururl.**

Appendix F How we selected our datasets

In order to select our datasets, the users need to download the datasets from authorized and reliable genome browsers like Ensembl, see link: http://www.ensembl.org/index.html, Biomart in the link bellow: http://www.biomart.org/ and GENCODE from this link: http://www.gencodegenes.org/.

How to make a dataset with sequences

The first thing users need to do is to make a dataset with all human genes with their FASTA format. To build our datasets, the users need to determine which types of genes they want to build in order to use them in the TIS tools. So let's see what is needed to create these datasets.

- Go to Ensembl genes browser and choose Biomart
- Choose Database
  - Ensembl Genes 80 or later versions
- Choose Dataset
  - Homo sapiens genes (GRCh38.p2)
- Attributes
  - Sequences
    - cDNA sequences
  - Header Information
    - Associated Gene Name
    - Ensembl Gene ID
    - Ensembl Transcript ID
    - CDS start (within cDNA)
    - CDS end (within cDNA)
- Results
  - Export all results to Compressed file (.gz)
  - Go

It will download a compressed file with 320-megabyte size. The compressed file will provide the dataset which the users will select genes from with their sequences in FASTA format. These sets will be tested to TIS Tools, CPAT, CPC, PLEK, PORTRAIT and OrfPredictor.

How to make the dataset "Coding genes 11.955" from BIOMART

In this section we will introduce the way to make the dataset for the coding genes, known as positive set. The first thing the users need to do is to visit the gene browser Biomart and take the following steps:

- Visit Biomart genes browser and choose browser data
- In the Datasets tab choose
  - Database:
    - Ensembl 79 Genes(WTSI, UK)
  - Datasets:
    - Homo sapiens genes(GRCh38.p2)
- In the Filters tab, do not pick go to the Output tab
- In the Output tab choose
  - Attributes
    - Features, Gene, Ensembl
      - Version(Transcript)
      - Status(Transcript)
      - Status(Gene)
      - Transcript type
      - Gene type
      - Associated Gene Name
      - GENCODE basic annotation
      - Transcript Support Level(TSL)
      - Transcript Length
      - Transcript End(bp)
      - Transcript Start(bp)
      - Strand
      - Chromosome Name
      - Ensembl Protein ID
      - Ensembl Transcript ID
      - Ensembl Gene ID
    - External References
      - UniProt/SwissProt ID
      - RefSeq mRNA[e.g.NM_001195597]
      - CCDS ID
      - VEGA transcript ID(s) (OTTT)
      - APPRIS
    - Results
      - Download Data

The Biomart will return a text file named "results.txt" with size 35.9 MB.

How to filter the dataset "Coding genes 11.955" from BIOMART

Once we have finished with our dataset, the next step is to filter it. So, let's see how to filter our dataset. To do this, users need to open the dataset as .xlsx. and choose:

- UniProt/SwissProt ID
  - (Select All / No Blanks)
- RefSeq mRNA [e.g.NM_001195597]
  - (Select All / No Blank)
- CCDS ID
  - (Select All / No Blanks)
- Version(transcript)
  - (Select All)
- Status(transcript)
  - (KNOWN)
- Status(gene)
  - (KNOWN)
- Transcript type
  - (protein_coding)
- Gene type
  - (protein_coding)
- Associated Gene Name
  - (Select All)
- APPRIS principal isoform annotation
  - (Select All / No Blanks)
- GENCODE basic annotation
  - (Select All)
- Transcript Support Level(TSL)
  - (tsl1-3 / No Blanks / No tsl4, tsl5, tslNA)
- VEGA transcript ID(s)(OTTT)
  - (All / No Blanks)

The attributes Transcript length, Transcript End (bp), Transcript Start (bp), Strand, Chromosome Name, Ensembl Protein ID, Ensembl Transcript ID and Ensembl Gene ID did not need filtering. So, the filtered dataset returned 14.252 genes.

Why do we need to keep the unique Coding genes from the dataset "Coding genes 11.955" from BIOMART?

In this section we will discuss the reason why we need to keep the unique coding genes from the filtered dataset. The reason we need to keep the longest mRNA for each gene is that some genes have only one transcript mRNA and some others can have more than one, because of exons and their combinations. So, to achieve high accuracy in our prediction we need to keep the longest mRNA transcript for each gene. Now the users must choose three attributes from the dataset:

- Associated Gene Name
- Ensembl Transcript ID
- Transcript length
- Choose Data and Sort the tables like:
  - Associated Gene Name
    - A-Z
  - Transcript length
    - Large to Small
- Save the document as:
  - Excel Workbook
  - Text(tab)
  - CSV

After that, we take the new text document and, with a script program, remove the duplicated genes and keep the longest mRNA transcript for each gene. The genes which remained were 11.955 in total.

How to make the dataset "Ensembl-Biomart & GENCODE"

To be sure that we will pick the most common non-coding genes, we have to make a dataset with all genes from two different bases, Ensembl-Biomart and GENCODE. The result is a dataset with seventeen attributes from both bases and totally 214.286 genes. The attributes are:

- Ensembl Transcript ID
- Ensembl Gene ID
- Ensembl Description
- Ensembl Gene Start
- Ensembl Gene End
- Ensembl Transcript Length
- Ensembl Transcript Support Level
- Ensembl GENCODE basic annotation
- Ensembl Associated Gene Name
- Ensembl Gene Type

- Ensembl Transcript Type
- Ensembl RefSeq NR
- GENCODE Transcript ID
- GENCODE Gene Name
- GENCODE Gene Status
- GENCODE Transcript Status
- GENCODE Transcript Support Level

How to make the dataset "Long non-coding genes 6.250" from Ensembl-Biomart & GENCODE

The purpose of this section is to present how to pick the most common long non-coding genes from the above dataset, "Ensembl-Biomart & GENCODE". The first thing one must do is to filter the dataset which returns 11.140 genes in total. But after that, we need to remove the duplicated genes. The dataset was filtered, considering the attributes below:

- Ensembl Gene Type
  - lncRNA
- Ensembl Transcript Type
  - lncRNA
- GENCODE Transcript Status
  - KNOWN

Why we need to remove duplicate genes from our dataset "Long non-coding genes 6.250" from Ensembl-Biomart & GENCODE

The reason why we need to remove the duplicate genes is the fact that we want to pick the long non-coding genes which are unique. The reason is because we need to have the most accurate results possible for the prediction. The users must choose three attributes from the dataset:

- Associated Gene Name
- Ensembl Transcript ID
- Transcript length
- Choose Data and Sort the tables like:
  - Associated Gene Name
    - A-Z
  - Transcript length

- Large to Small

- Save the document as:
  - Excel Workbook
  - Text(tab)
  - CSV

After that, we take the new text document and, with a script program, remove the duplicate genes and keep the unique long non-coding genes. The genes which remain are totally 6.250. These are the genes which are used for our prediction.

Appendix G Results from TIS Tools for the sets

Introduction

In this chapter our aim is to present and compare the results of the two datasets, the dataset for coding genes, "Coding genes 11.955", and the dataset for long non-coding genes, "Long non-coding genes 6.250", run with the previously mentioned programs.

"Coding genes 11.955" TIS Tools Results, Differences & Comparison

In this section we will present the results, the differences and the comparisons for the dataset "Coding genes 11.955" for TIS Tools, such as CPAT, CPC, PLEK, PORTRAIT and OrfPredictor.

CPAT results for dataset "Coding genes 11.955"

CPAT returned excellent results for the prediction of Coding genes with a percentage of 97.7% accuracy (see Figure 5.2). CPAT returns three files, "output2", "output2.dat" and "output2.r" after finishing with the prediction. The most significant file was the "output2" which contains six attributes. The attributes were: "mRNA size", "ORF size", "Transcript length", "Fickett score", "Hexamer score" and "Coding probability". An important detail is that CPAT detects as coding genes those with probability score equal to >= 0,364. Example of CPAT results (see Table appendix 1.1):

mRNA size	*ORF size*	*Transcript length*	*Fickett score*	*Hexamer score*	*Coding probability*
LLGL1 | ENSG00000131	3195	4260	11971	0.5210719152	1
LEPR | ENSG000001166	3498	8227	0.6382	-0.047639341	1
SMIM15 | ENSG000001	261	2889	0.6733	-0.503522132	0.003014182
OST4 | ENSG000002284	117	561	0.8146	-0.262144065	0.004806479
C14ORF2 | ENSG00000	177	643	0.8474	0.0945008133	0.084022062
COX20 | ENSG0000020	357	2631	0.6056	-0.075859121	0.086952384

UBL3 | ENSG000001220	354	4384	0.7674	-0.008322209	0.180416918
COA5 | ENSG00000183	225	1754	0.8589	0.1546023877	0.181167149
MRPL33 | ENSG000002	198	518	0.9553	0.1411021336	0.183565434
COPS2 | ENSG0000016	1332	6628	0.8938	0.1229986903	0.999973483

Table appendix 1.1: Example of CPAT results for coding genes. This table has been designed for the purpose of this thesis.

CPAT returns the "ORF size". That means that it can predict the Open Reading Frame (ORF) for each gene. Only CPAT and PORTRAIT gave users this information. CPAT detects the "ORF size" with 97.36% success rate (see Figure 5.1). To make this comparison we made a program which compared the "CDS length" from Ensembl and the "ORF size" of CPAT output for each gene. The results for this comparison are shown in a new tab named "Equal/Not Equal". That shows the association between "CDS length" and "ORF size". Example for CPAT how it predicted ORF size (see Table appendix 1.2):

mRNA size	***CDS length***	***ORF size***	***Equal/Not Equal***	***Transcript length***	***Fickett score***	***Hexamer score***	***Coding probability***
LLGL1 | 3195	3195	3195	EQUAL	4260	11971	0.52107	1
LEPR | 3498	3498	3498	EQUAL	8227	0.6382	-0.04763934	1
RIMS1 | 5079	5079	5079	EQUAL	5079	0.7024	0.160987573	1
UPF1 | 3357	3357	3357	EQUAL	5348	12715	0.564062099	1
ZNF197 | 3090	3090	3090	EQUAL	3275	0.8835	0.120634713	1
CENPF | 9345	9345	9345	EQUAL	10307	10975	0.185642958	1
SYNJ2 | 4491	4491	4491	EQUAL	7378	10549	0.356128968	1
PXDNL | 4392	4392	4392	EQUAL	4805	0.8436	0.202934828	1
MTR | 3798	3798	3798	EQUAL	10529	11808	0.247650143	1
PSMD2 | 2727	2727	2967	NOT_EQUAL	3449	11711	0.378994377	1

Table appendix 1.2: Example of CPAT how predict ORF size. This table has been designed for the purpose of this thesis.

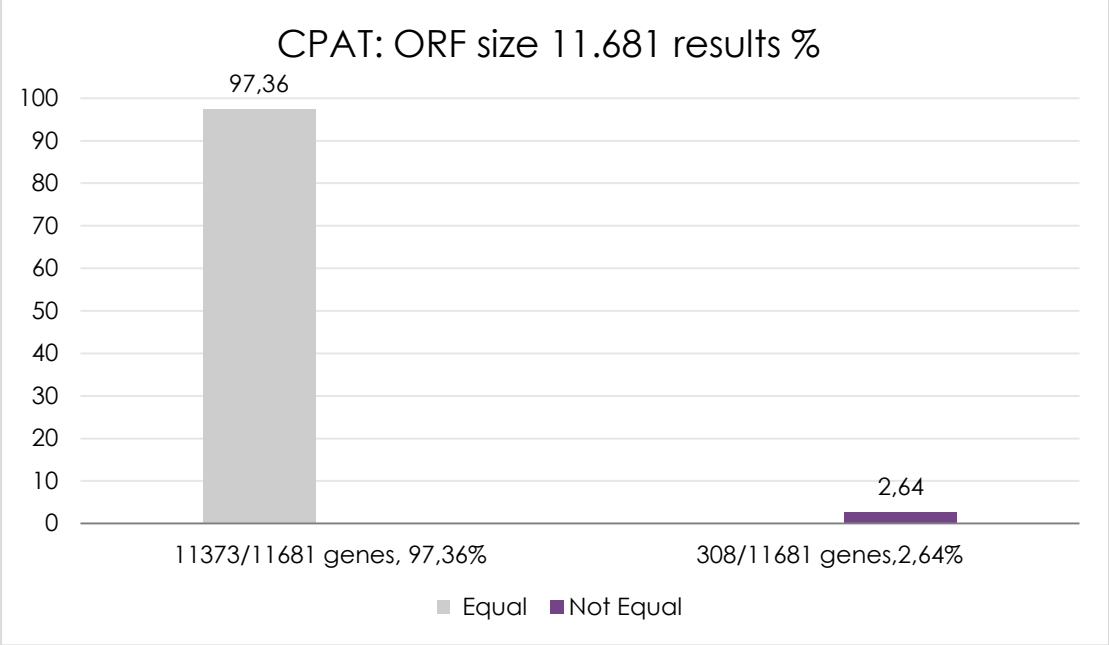


Figure appendix 1.1: CPAT results for ORF size of dataset "Coding genes 11.681". This figure has been designed for the purpose of this thesis.

CPAT predicted as coding 11.681 genes out of 11.955 coding genes with 97.7% success rate and only 274 coding genes detected as non-coding with 2.3% success rate.

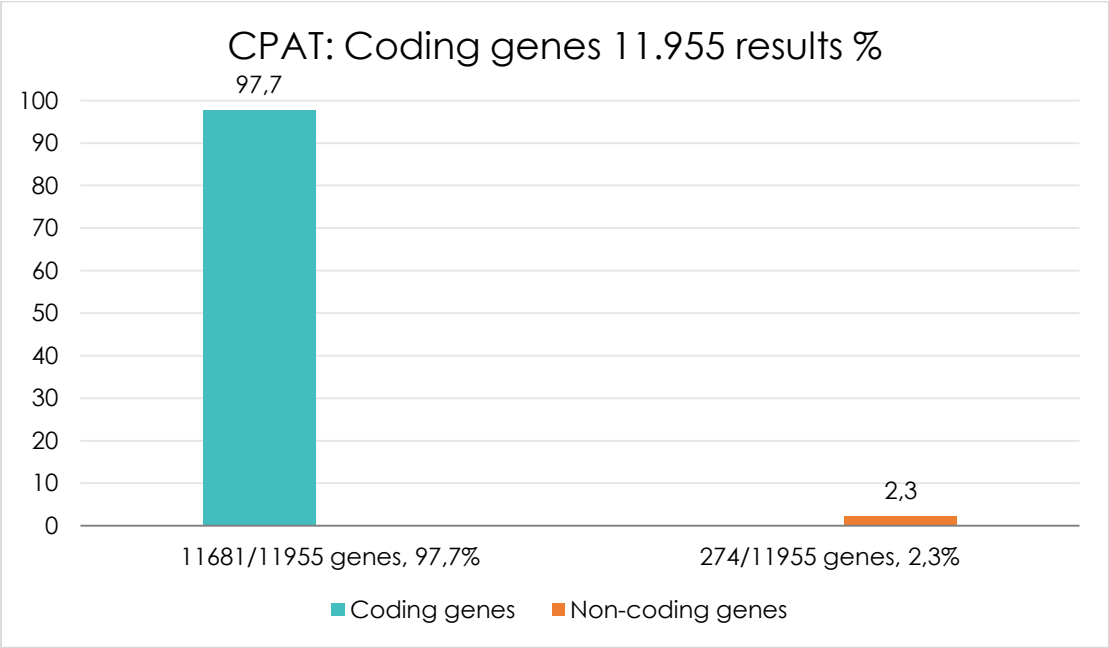


Figure appendix 1.2: CPAT results for dataset "Coding genes 11.955". This figure has been designed for the purpose of this thesis.

CPC results for dataset "Coding genes 11.955"

CPC results for the prediction of coding genes achieved 85.1% success rate. CPC printed three documents, "result_evidence.HOMO" file, "result_evidence.ORF" file and "result_in_table", after finishing with the prediction. The file "result_in_table" was the most significant of the three files and contained four features: "mRNA", "Transcript length", "Coding/Non Coding" and "Coding probability". The other two files contained information about the "BLASTX". The "result_evidence.ORF" file contained information's for "ORF frame" and only for the coding genes. Example of CPC results:

mRNA	***Transcript length***	***Coding/NonCoding***	***Coding Probability***
YIPF3 | ENSG00000137207	1576	coding	220.951
LLGL1 | ENSG00000131899	4260	coding	784.909
LEPR | ENSG00000116678	8227	coding	426.735
DAPK2 | ENSG00000035664	1741	coding	244.912
GLA | ENSG00000102393	1318	coding	133.811
PLA2G10 | ENSG00000069764	1000	noncoding	-0.448982
NDUFC1 | ENSG00000109390	809	noncoding	-0.963918
KLHDC9 | ENSG00000162755	1314	noncoding	-0.249919
PTS | ENSG00000150787	935	noncoding	-0.311758
SERF2 | ENSG00000140264	1931	noncoding	-0.876614

Table appendix 1.3: Example of CPC results for coding genes. This table has been designed for the purpose of this thesis.

CPC predicted as coding the 10.171 out of the 11.955 coding genes with 85.1% success rate, while 1.784 coding genes were detected as non-coding with 14.9% false rate. (see Figure appendix 1.3)

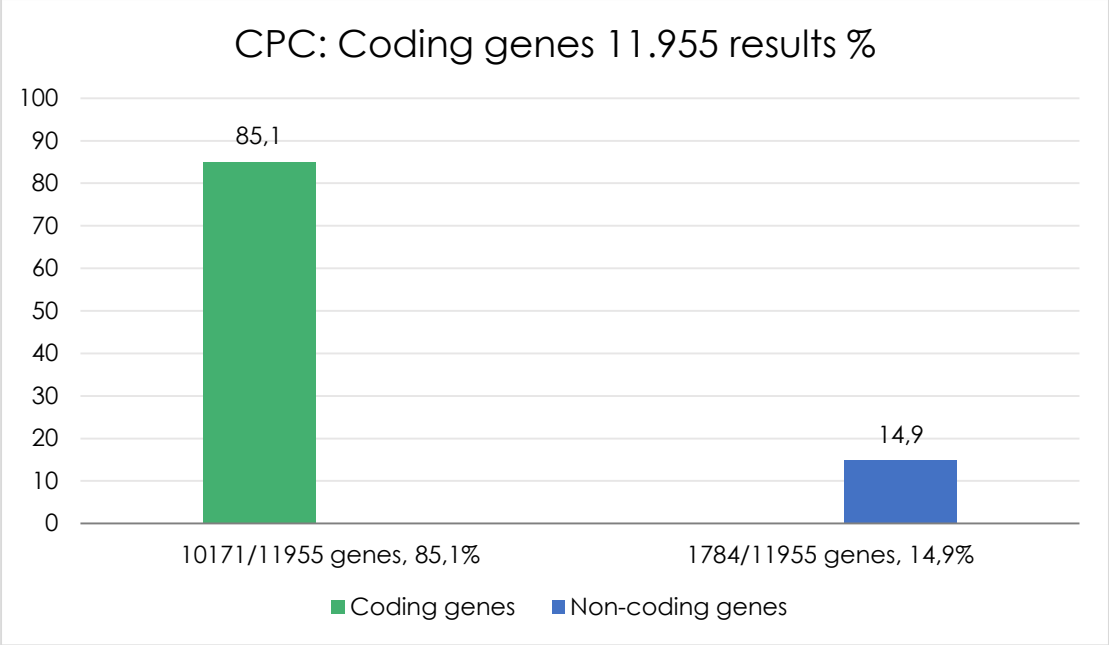


Figure appendix 1.3: CPC results for dataset "Coding genes 11.955". This figure has been designed for the purpose of this thesis.

PLEK results for dataset "Coding genes 11.955"

PLEK results for the prediction of coding genes had 93.4% success rate (see Figure 5.6). PLEK printed one file after finishing with the prediction, named "Predicted". The file contained three features, "mRNA", "Coding/Non Coding" and "Coding probability". PLEK did not return the "ORF size". Example of PLEK results (see Table appendix 1.4):

mRNA	***Coding/Non-coding***	***Coding Propability***
YIPF3 | ENSG00000137207	Coding	1.731.320
IL20 | ENSG00000162891	Coding	1.105.280
LLGL1 | ENSG00000131899	Coding	1.867.890
RPS6KB2 | ENSG00000175634	Coding	1.930.390
MXD4 | ENSG00000123933	Coding	1.232.050
LYSMD1 | ENSG00000163155	Non-coding	-0.008743
BPIFA2 | ENSG00000131050	Non-coding	-0.869155
IL36A | ENSG00000136694	Non-coding	-0.583599
PCYT1B | ENSG00000102230	Non-coding	-0.027357
SERPINA6 | ENSG00000170099	Non-coding	-0.096317

Table appendix 1.4: Example of PLEK results for coding genes. This table has been designed for the purpose of this thesis.

PLEK predicted as coding 11.165 out of 11.955 coding genes with 93.4% success rate and 790 coding genes were detected as non-coding with 6.6% false rate. (see Figure appendix 1.4)

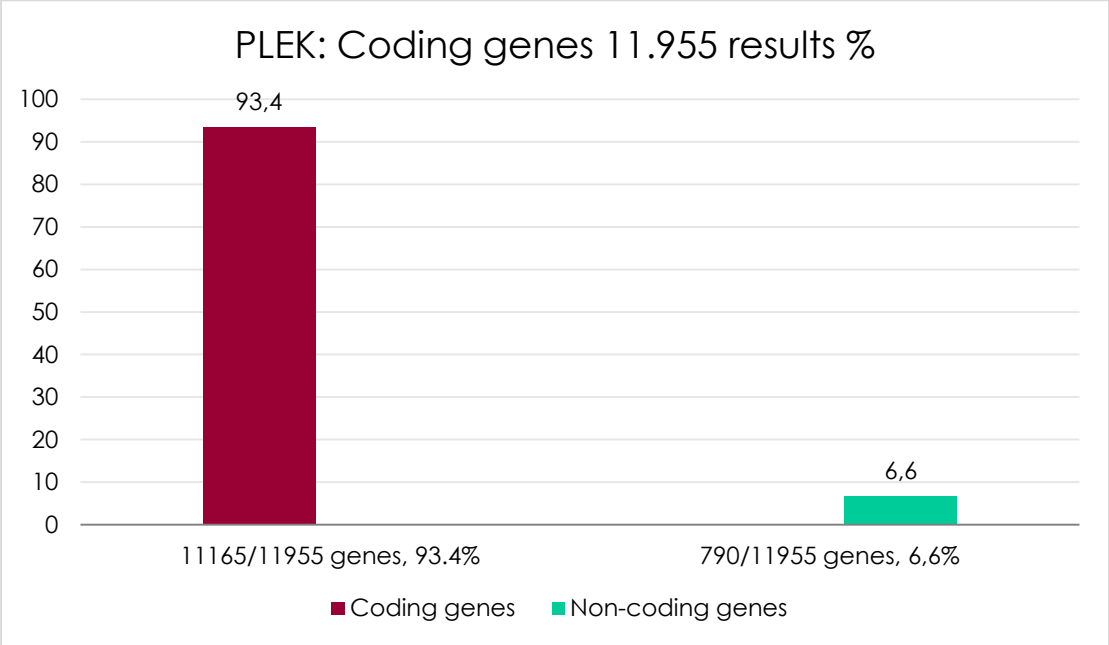


Figure appendix 1.4: PLEK results for dataset "Coding genes 11.955". This figure has been designed for the purpose of this thesis.

PORTRAIT results for dataset "Coding genes 11.955"

PORTRAIT results for the prediction of coding genes had 99.7% success rate. PORTRAIT printed three to sixteen text files after finishing with the prediction. The files were: "Name.FASTA", "Name.FASTA_RESULTS", "Name.ORF", "Name.CAST_PARSED", "Name.CAST_RAW", "Name.AACOMP", "Name.IEP", "Name.ORFSIZE", "Name.SOAP", "Name.WITHORFS", "Name.NT1", "Name.NT2", "Name.NT3", "Name.RNASIZE", "Name.WITHOUTORFS". We took the "ORF file" which contained two features, "mRNA" and "ORF size". Example of PORTRAIT results (see Table appendix 1.5):

mRNA	***ORF size***
YIPF3 | ENSG00000137207	1200
IL20 | ENSG00000162891	531
LLGL1 | ENSG00000131899	3291
RPS6KB2 | ENSG00000175634	1494
MXD4 | ENSG00000123933	798
C1orf100 | ENSG00000173728	N/A
DEFB105B | ENSG00000186599	N/A
LMO7DN | ENSG00000178734	N/A

USMG5 | ENSG00000173915	N/A
DEFB124 | ENSG00000180383	N/A

Table appendix 1.5: Example of PORTAIT results for coding genes. This table has been designed for the purpose of this thesis.

PORTRAIT returned the "ORF size". That means that it can predict the Open Reading Frame (ORF) for each gene (see Figure 5.5). But to make the comparison more accurate, we made a program which compared "CDS length" from Ensembl and the "ORF size" of PORTRAIT output for each gene. The program returned a new tab named "Equal/Not Equal". This tab showed association between "CDS length" and "ORF size" for each gene of the set. PORTRAIT returned the "ORF size" for the predicted coding genes with 52.63% percentage of success rate as Equal and the 47.37% rate as Not Equal. Example for PORTRAIT how predicted "ORF size" (see Table appendix 1.6):

mRNA	***CDS Length***	***ORF Size***	***EQUAL/NOT EQUAL***
YIPF3 | ENSG00000137207	1053	1200	NOT_EQUAL
IL20 | ENSG00000162891	531	531	EQUAL
LLGL1 | ENSG00000131899	3195	3291	NOT_EQUAL
RPS6KB2 | ENSG00000175634	1449	1494	NOT_EQUAL
MXD4 | ENSG00000123933	630	798	NOT_EQUAL
LEPR | ENSG00000116678	3498	3291	NOT_EQUAL
DAPK2 | ENSG00000035664	1113	1113	EQUAL
GLA | ENSG00000102393	1290	1227	NOT_EQUAL
FCGRT | ENSG00000104870	1098	1098	EQUAL
PLBD1 | ENSG00000121316	1662	1662	EQUAL

Table appendix 1.6: Example of PORTRAIT how predict ORF size results. This table has been designed for the purpose of this thesis.

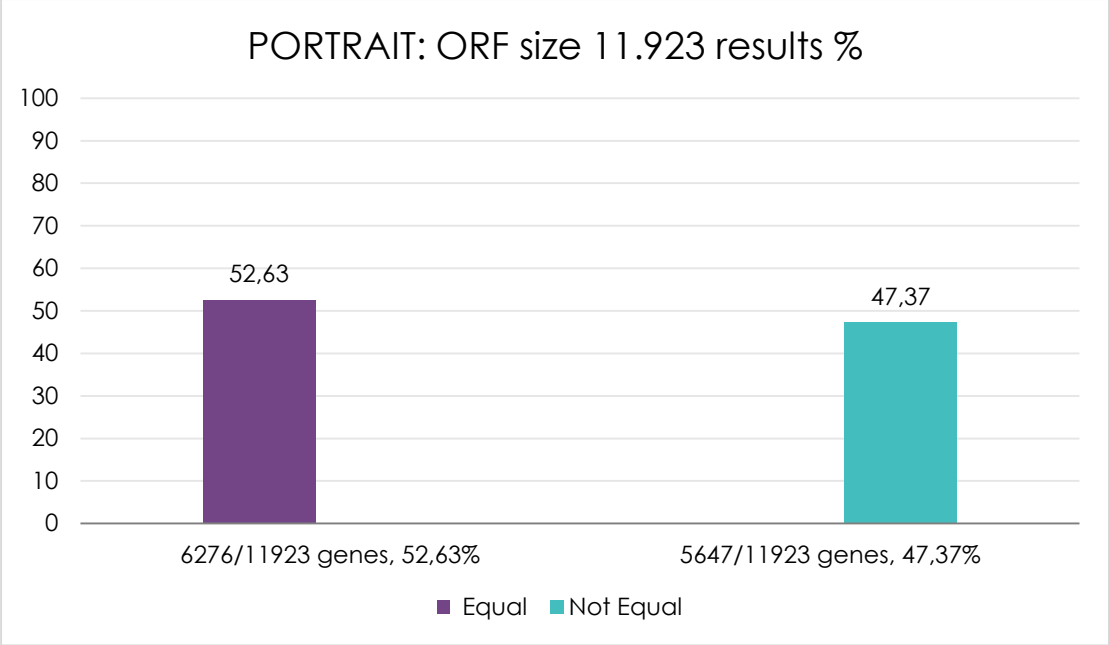


Figure appendix 1.5: PORTRAIT results for ORF size of dataset "Coding genes 11.681". This figure has been designed for the purpose of this thesis.

PORTRAIT predicted as coding 11.923 out of 11.955 coding genes with 99.7% success rate and only 32 coding genes were predicted as non-coding with 0.3% false rate.

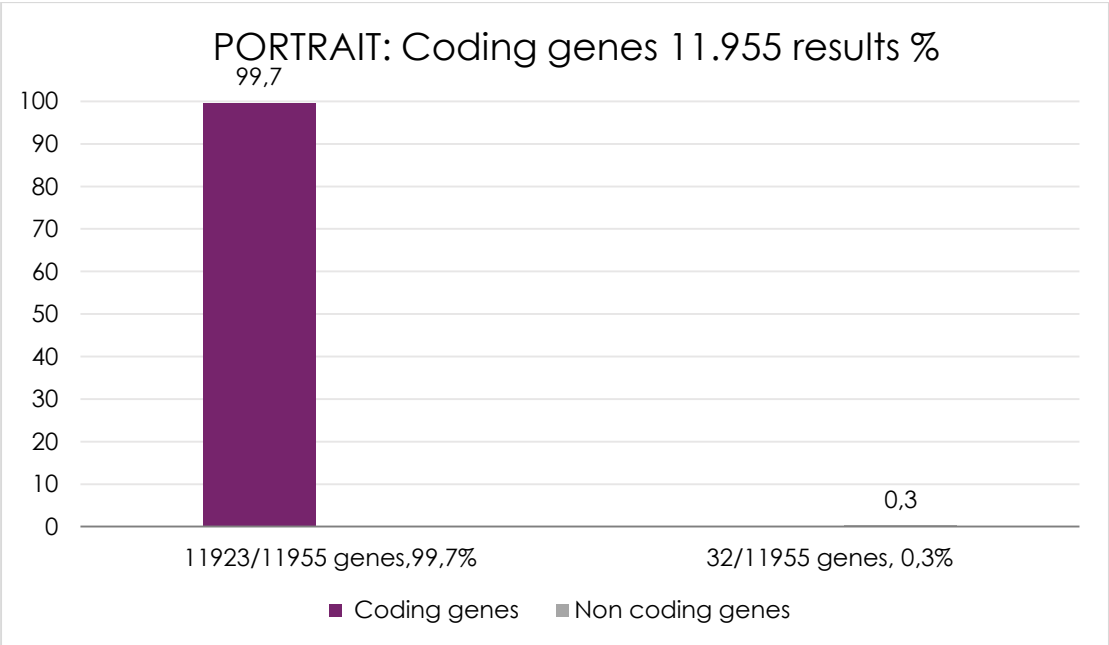


Figure appendix 1.6: PORTRAIT results for dataset "Coding genes 11.955". This figure has been designed for the purpose of this thesis.

OrfPredictor results for dataset "Coding genes 11.955"

The OrfPredictor results for the prediction of coding genes were 100% accurate(see Figure 5.7). OrfPredictor printed three files after finishing with the prediction, "noORF", "ORF6frame" and "output". Example of OrfPredictor from file "output", showed in table below (see Table appendix 1.7):

mRNA	*Strand*	*ORF end*	*ORF start*
YIPF3 | ENSG00000137207	1	184	1233
IL20 | ENSG00000162891	3	45	572
LLGL1 | ENSG00000131899	1	97	3288
RPS6KB2 | ENSG00000175634	1	46	1491
MXD4 | ENSG00000123933	3	147	941
LEPR | ENSG00000116678	3	186	3680
DAPK2 | ENSG00000035664	2	32	1141
GLA | ENSG00000102393	2	23	1309
FCGRT | ENSG00000104870	1	487	1581
PLBD1 | ENSG00000121316	3	654	2312

Table appendix 1.7: Example of OrfPredictor results for coding genes. This table has been designed for the purpose of this thesis.

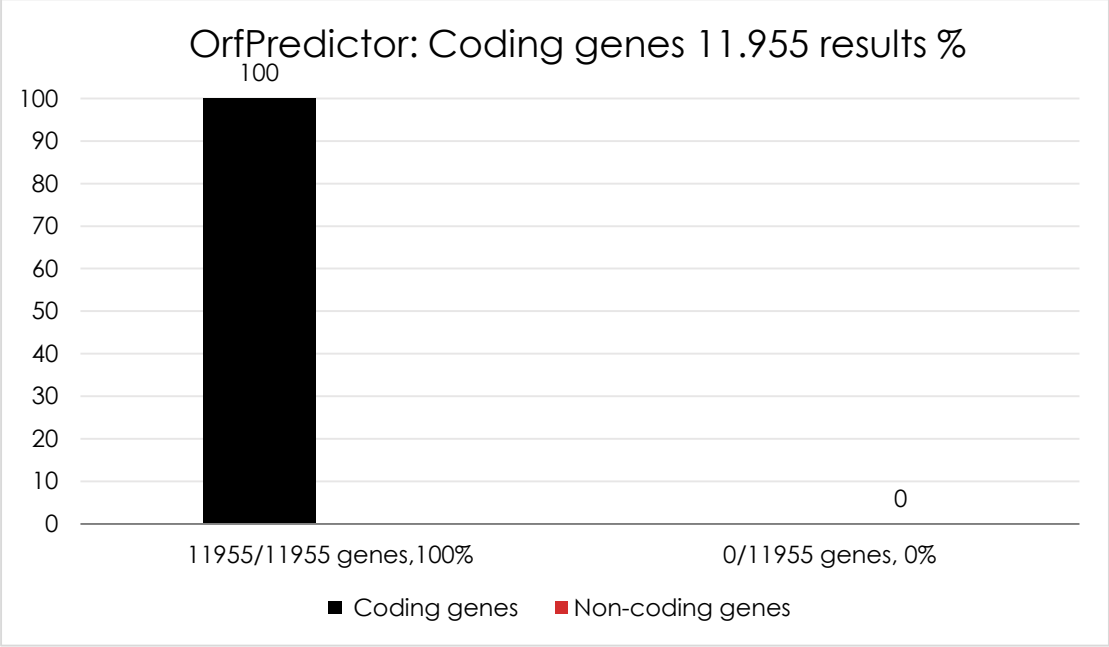


Figure appendix 1.7: OrfPredictor results for dataset "Coding genes 11.955". This figure has been designed for the purpose of this thesis.

All TIS Tools results for dataset "Coding genes 11.955"

All the results from the TIS Tools are presented in the chart below. As we can see all tools have gathered very good results (see Figure appendix 1.8 & Figure appendix 1.9).

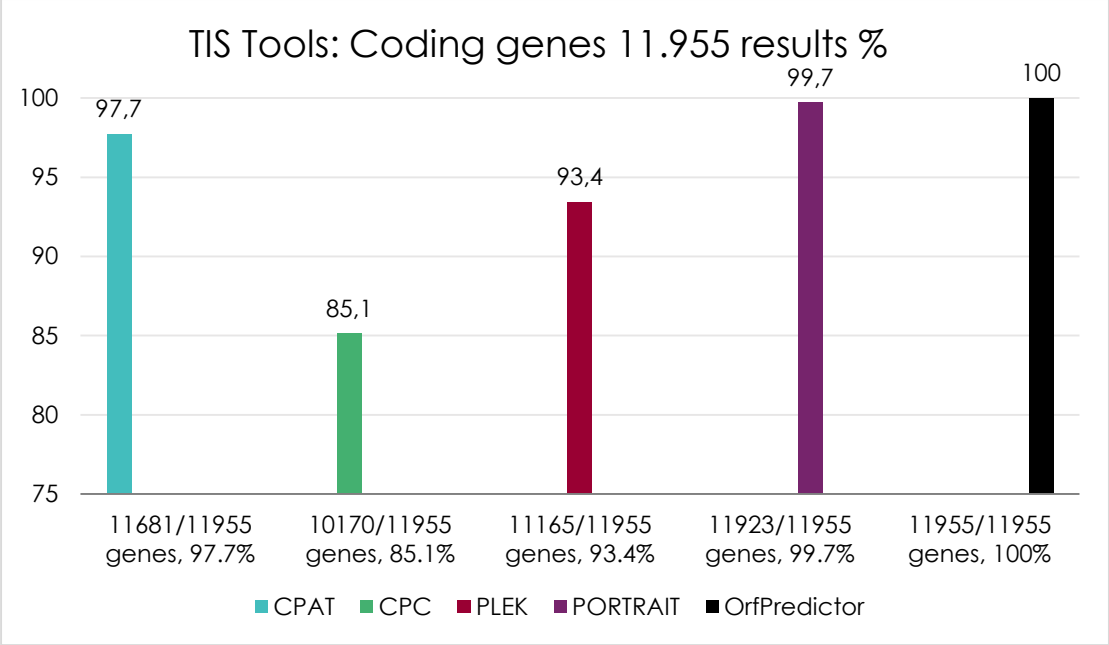


Figure appendix 1.8: TIS Tools results for dataset "Coding genes 11.955". This figure has been designed for the purpose of this thesis.

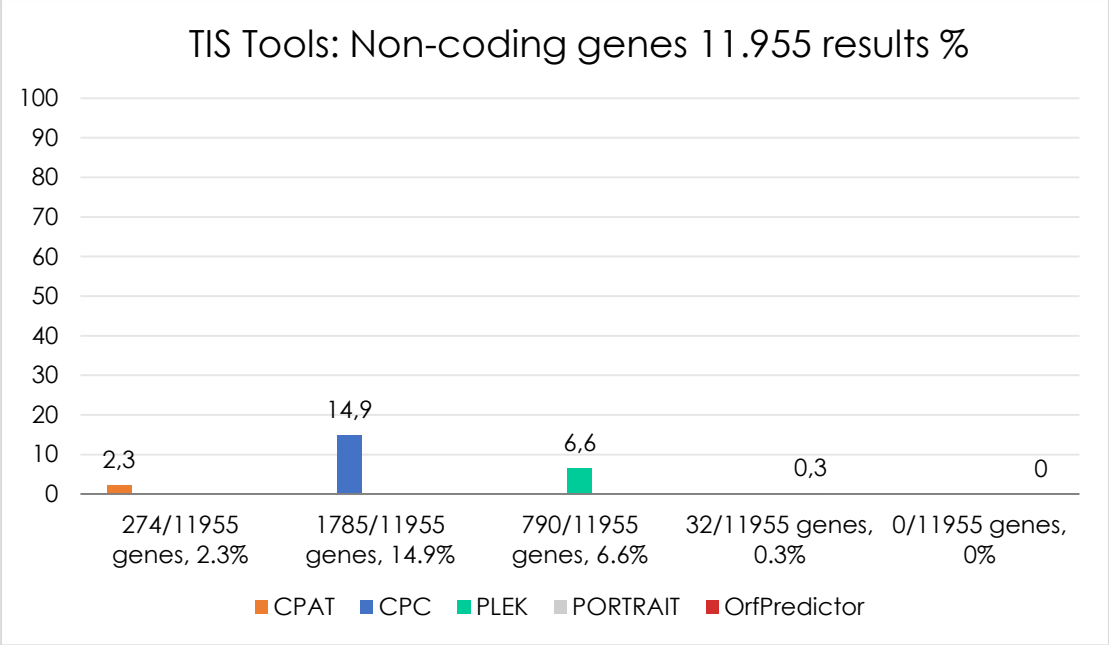


Figure appendix 1.9: Tools results Non-coding for dataset "Coding genes 11.955". This figure has been designed for the purpose of this thesis.

"Long non-coding genes 6.250" TIS Tools Results, Differences & Comparison

In this section we will present the results, the differences and the comparisons for the dataset "Long non-coding genes 6.250" for TIS Tools, such as CPAT, CPC, PLEK, PORTRAIT and OrfPredictor.

CPAT results for dataset "Long non-coding genes 6.250"

CPAT returned very good results for the prediction of Long non-coding genes with a percentage of 93.5% accuracy(see Figure 5.10). CPAT returns three files, "output2", "output2.dat" and "output2.r" after finishing with the prediction. The most significant file was the "output2" which contains six attributes. The attributes were: "mRNA size", "ORF size", "Transcript length", "Fickett score", "Hexamer score" and "Coding probability". An important detail is that CPAT detects as coding genes those with probability score equal to >= 0,364. Example of CPAT results (see Table appendix 1.8):

mRNA size	ORF size start	ORF size end	Fickett score	Hexamer score	Coding probability
PCAT29 | ENSG00000259641	201	2121	0.6425	-0.061896285	0.020697418
LINC01493 | ENSG00000254562	168	581	0.8389	-0.274950698	0.008514461
CTD-2313F11.3 | ENSG00000248029	159	567	0.8128	-0.120467217	0.017812465
RP11-19P22.5 | ENSG00000265788	354	2110	0.5025	-0.012582804	0.088842652
RP11-763E3.1 | ENSG00000262052	162	585	0.9535	-0.141441226	0.026095088
ZNF833P | ENSG00000197332	3383	837	0.909	-0.099152971	0.979806783
ZNF883 | ENSG00000228623	2319	1044	1.0068	0.058714362	0.999474872
U73166.2 | ENSG00000230454	2625	495	0.6333	0.013510196	0.457757046
TUNAR | ENSG00000250366	3197	198	1.1488	0.212493369	0.368415692
TRBV11-2 | ENSG00000241657	408	345	0.8835	0.157222304	0.502144887

Table appendix 1.8: Example of CPAT results for long non-coding genes. This table has been designed for the purpose of this thesis.

CPAT predicted as long non-coding 5.847 out of 6.250 long non-coding genes with 93.5% success rate and only 403 long non-coding genes detected as coding with 6.5% false rate.

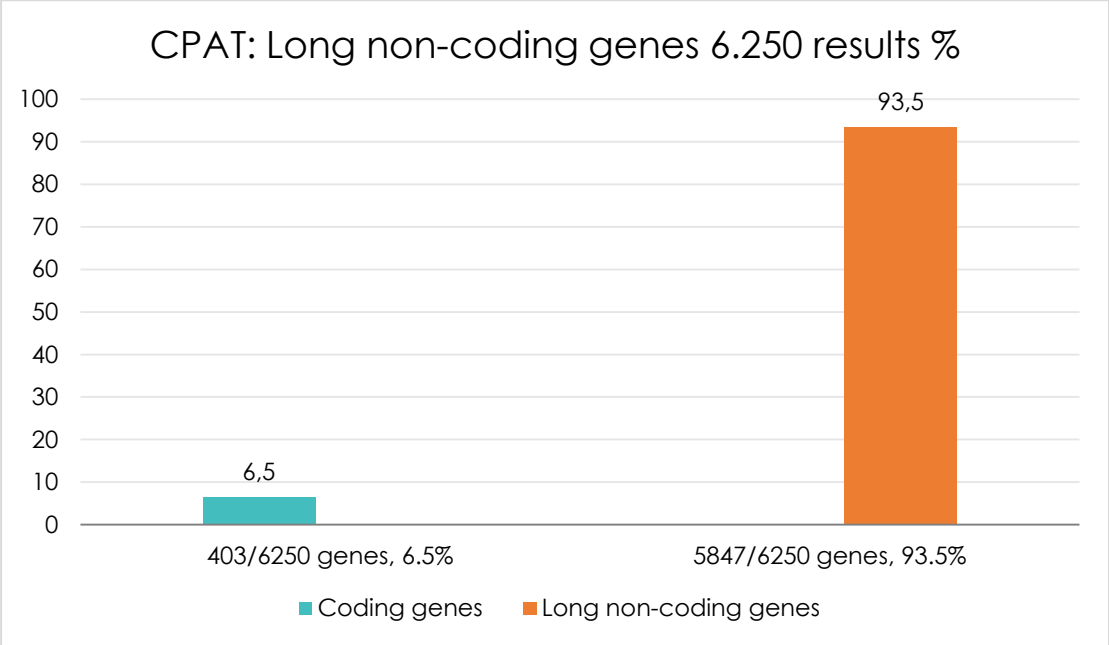


Figure appendix 1.10: CPAT results for dataset "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

CPC results for dataset "Long non-coding genes 6.250"

CPC results for the prediction of long non-coding genes achieved 98.8% success rate (see Figure 5.11). CPC printed three documents, "result_evidence.HOMO" file, "result_evidence.ORF" file and "result_in_table", after finishing with the prediction. The file "result_in_table" was the most significant of the three files and contained four features: "mRNA", "Transcript length", "Coding/Non Coding" and "Coding probability". The other two files contained information about the "BLASTX". The "result_evidence.ORF" file contained information's for "ORF frame" and only for the coding genes. Example of CPC results: (see Table appendix 1.9):

mRNA	***Transcript length***	***Coding/Non Coding***	***Coding Probability***
PCAT29 | ENSG00000259641	2121	noncoding	-1.12625
LINC01493 | ENSG00000254562	581	noncoding	-1.08397
CTD-2313F11.3 | ENSG00000248029	567	noncoding	-0.779052
RP11-19P22.5 | ENSG00000265788	2110	noncoding	-0.898423
RP11-763E3.1 | ENSG00000262052	585	noncoding	-0.913101
C10orf91 | ENSG00000180066	1846	coding	0.0566741
AP000696.2 | ENSG00000231324	1133	coding	0.178134
CTC-786C10.1 | ENSG00000262601	777	coding	0.330483
LINC00371 | ENSG00000226792	2743	coding	0.370696
LINC00940 | ENSG00000235049	2350	coding	1.12195

Table appendix 1.9: Example of CPC results for long non-coding genes. This table has been designed for the purpose of this thesis.

CPC predicted as long non-coding the 6.176 out of 6.250 of long non-coding genes with 98.8% success rate, while 74 long non-coding genes were detected as coding with 1.2% false rate. (see Figure appendix 1.11)

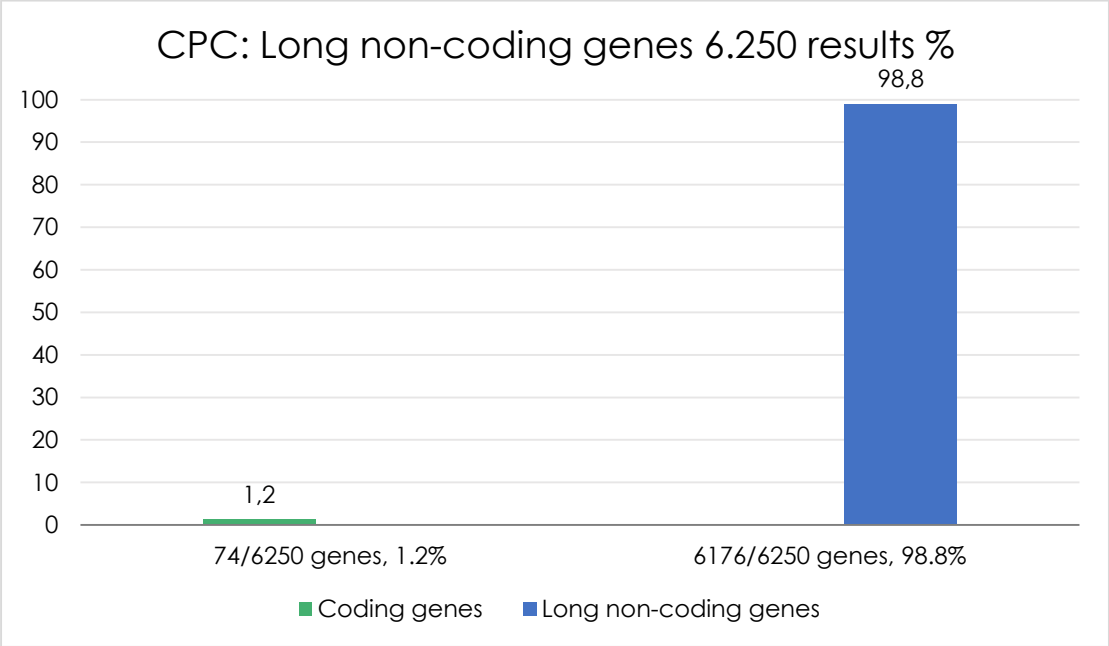


Figure appendix 1.11: CPC results for dataset "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

PLEK results for dataset "Long non-coding genes 6.250"

PLEK removed 63 genes as short genes from 6.250 long non-coding genes. PLEK results for the prediction of long non-coding genes had 98% success rate (see Figure 5.12). PLEK printed one file after finishing with the prediction, named "Predicted". The file contained three features, "mRNA","Coding/Non Coding" and "Coding probability". PLEK did not return the "ORF size". Example of PLEK result (see Table appendix 1.10):

mRNA	*Coding/Non-Coding*	*Coding probability*
LINC01012 | ENSG00000281706	Coding	0.348473
RP11-1102P16.1 | ENSG00000253379	Coding	0.410042
LINC00998 | ENSG00000214194	Coding	0.605475
RP11-403I13.5 | ENSG00000232721	Coding	0.10709
AC011747.4 | ENSG00000236008	Coding	0.264657
AC013727.1 | ENSG00000232597	Non-coding	-1.44806
RP4-754E20__A.5 | ENSG00000236117	Non-coding	-2.05644
LINC00678 | ENSG00000254934	Non-coding	-1.7675
RP13-463N16.6 | ENSG00000242147	Non-coding	-1.54915
RP11-35J1.2 | ENSG00000280511	Non-coding	-2.10556

Table appendix 1.10: Example of PLEK results for long non-coding genes. This table has been designed for the purpose of this thesis.

PLEK predicted as long non-coding 6.062 out of 6.187 long non-coding genes with 98% success rate and 2 long non-coding genes were detected as coding with 2% false rate.

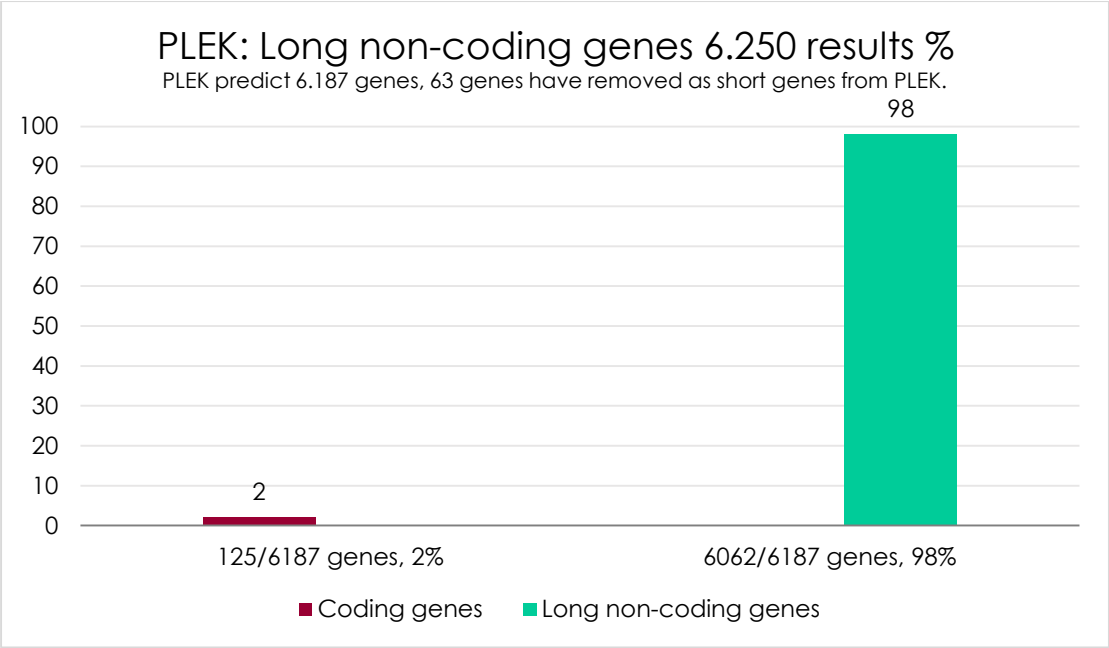


Figure appendix 1.12: PLEK results for dataset "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

PORTRAIT results for dataset "Long non-coding genes 6.250"

PORTRAIT results for the prediction of long non-coding genes had 42.4% success rate (see Figure 5.13). PORTRAIT printed three to sixteen text files after finishing with the prediction. The files were: "Name.FASTA", "Name.FASTA_RESULTS", "Name.ORF", "Name.CAST_PARSED", "Name.CAST_RAW", "Name.AACOMP", "Name.IEP", "Name.ORFSIZE", "Name.SOAP", "Name.WITHORFS", "Name.NT1", "Name.NT2", "Name.NT3", "Name.RNASIZE", "Name.WITHOUTORFS". We took the "ORF file" which contained two features, "mRNA" and "ORF size". Example of PORTRAIT results (see Table appendix 1.11):

mRNA	***Scores***
PCAT29 | ENSG00000259641	0.883721
LINC01493 | ENSG00000254562	0.861678
CTD-2313F11.3 | ENSG00000248029	0.564427

RP11-19P22.5 | ENSG00000265788	0.560353
RP11-111M22.5 | ENSG00000271757	0.932177
CTD-2231E14.5 | ENSG00000267373	0.949482
AP000654.4 | ENSG00000269895	0.730877
CTD-2308B18.3 | ENSG00000248296	0.812577
RP11-462L8.1 | ENSG00000229656	0.914486
RP11-37B2.1 | ENSG00000251136	0.720935

Table appendix 1.11: Example of PORTRAIT results for long non-coding genes. This table has been designed for the purpose of this thesis.

PORTRAIT predicted as long non-coding 2.651 out of 6.250 long non-coding genes with 42.4% success rate and 3.599 long non-coding genes were detected as coding with 57.6% false rate.

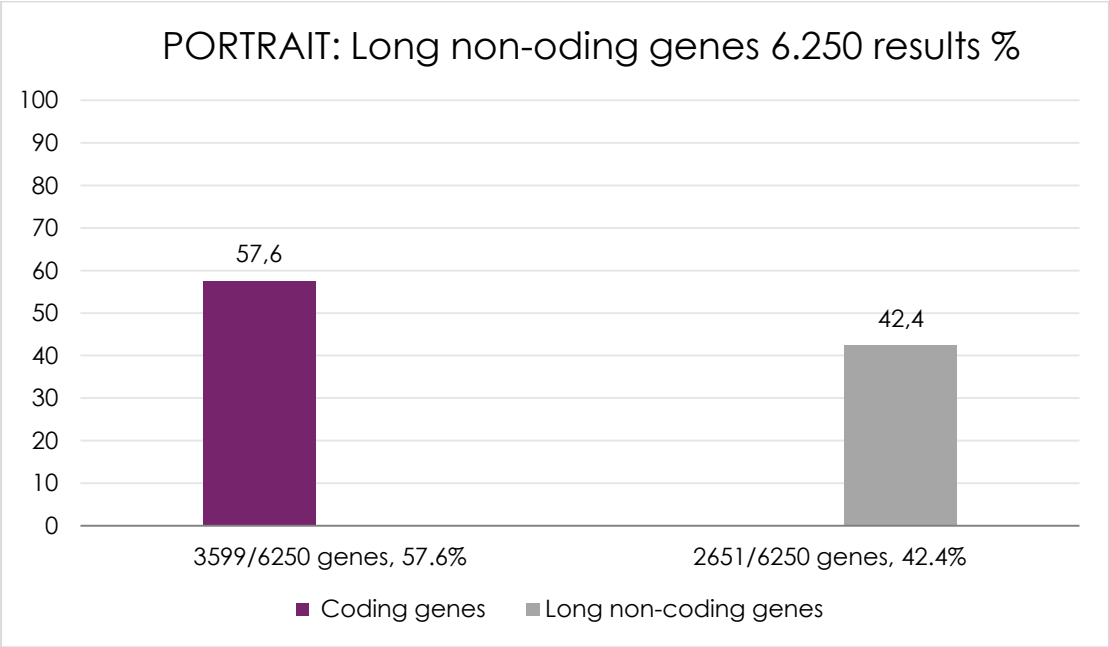


Figure appendix 1.13: PORTRAIT results for dataset "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

OrfPredictor results for dataset "Long non-coding genes 6.250"

The OrfPredictor results for the prediction of long non-coding genes had 1% success rate (see Figure 5.14). OrfPredictor printed three files after finishing with the prediction, "noORF", "ORF6frame" and "output". Example of OrfPredictor from file "output", showed in table below (see *Table appendix 1.12*).

mRNA	*Strand*	*ORF end*	*ORF start*
PCAT29 | ENSG00000259641	-1	4	264
LINC01493 | ENSG00000254562	1	265	429
CTD-2313F11.3 | ENSG00000248029 | ENST00000445660	-1	343	543
RP11-19P22.5 | ENSG00000265788 | ENST00000583179	1	28	378
RP11-763E3.1 | ENSG00000262052 | ENST00000571972	-2	227	583
RP11-111M22.5 | ENSG00000271757 | ENST00000607673	3	312	683
CTD-2231E14.5 | ENSG00000267373 | ENST00000587693	3	3	95
RP11-166A12.1 | ENSG00000251538 | ENST00000511194	-3	981	1181
CTD-2308B18.3 | ENSG00000248296 | ENST00000503145	1	1	369
AC008991.1 | ENSG00000267683 | ENST00000587850	-2	2	226

Table appendix 1.12: Example of OrfPredictor results for long non-coding genes. This table has been designed for the purpose of this thesis.

OrfPredictor predicted as long non-coding 12 out of 6.250 long non-coding genes with 1% success rate and 32 long non-coding genes were detected as coding with 99% false rate.

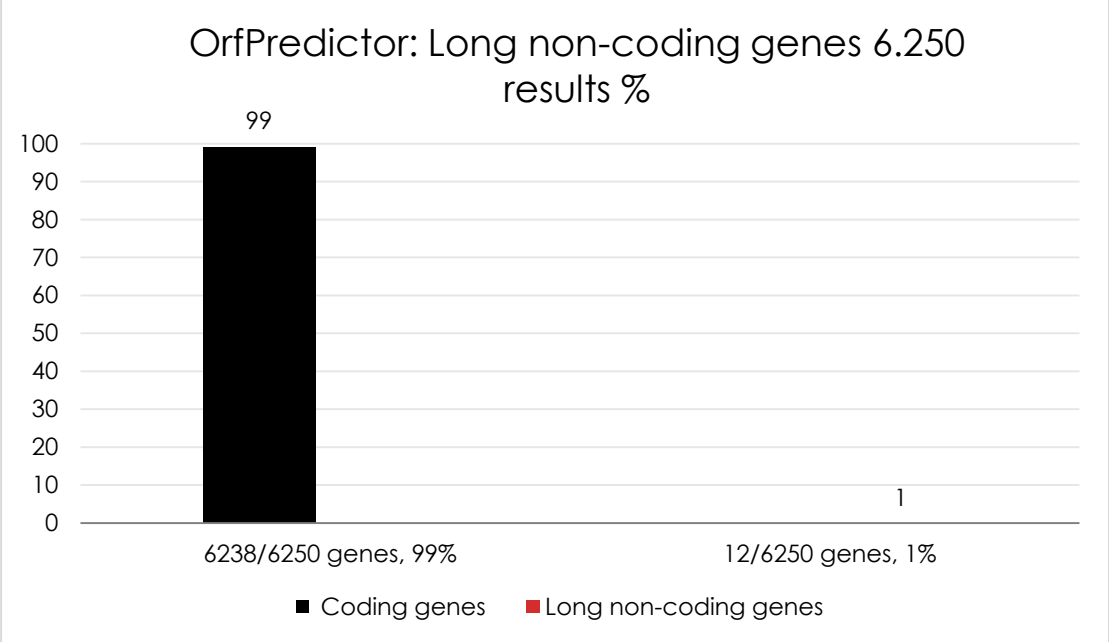


Figure appendix 1.14: OrfPredictor results for dataset "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

All TIS Tools results for dataset "Long non-coding genes 6.250"

All the results from TIS Tools are presented in the chart below. As we can see all tools have gathered very good results except OrfPredictor (see Figure appendix 1.15& Figure appendix 1.16).

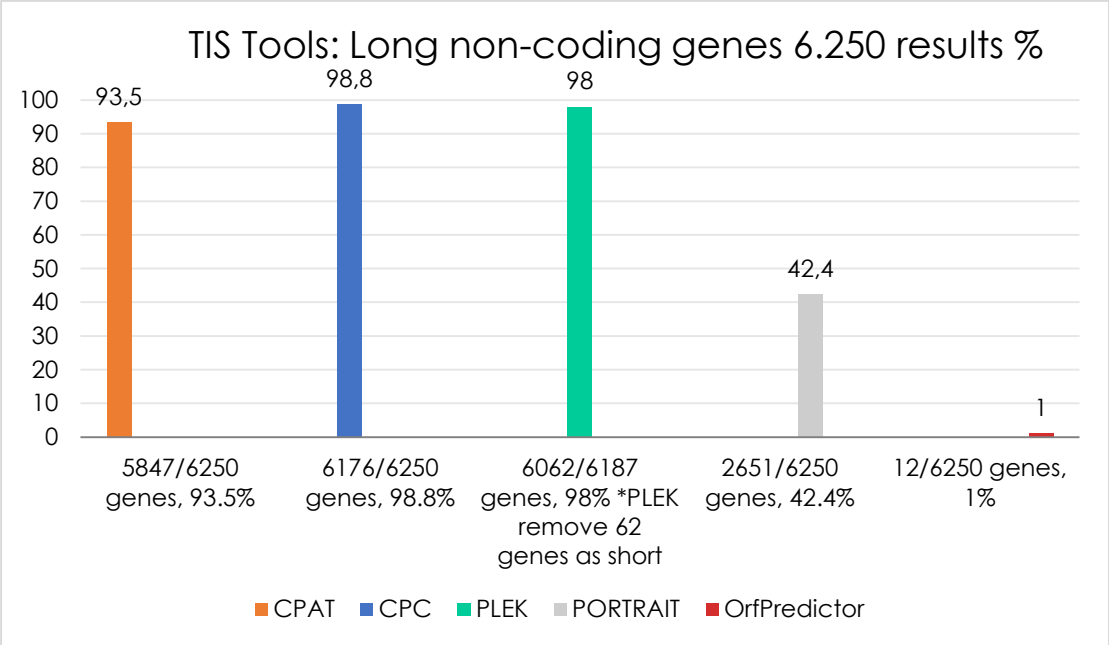


Figure appendix 1.15: TIS Tools results for dataset "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

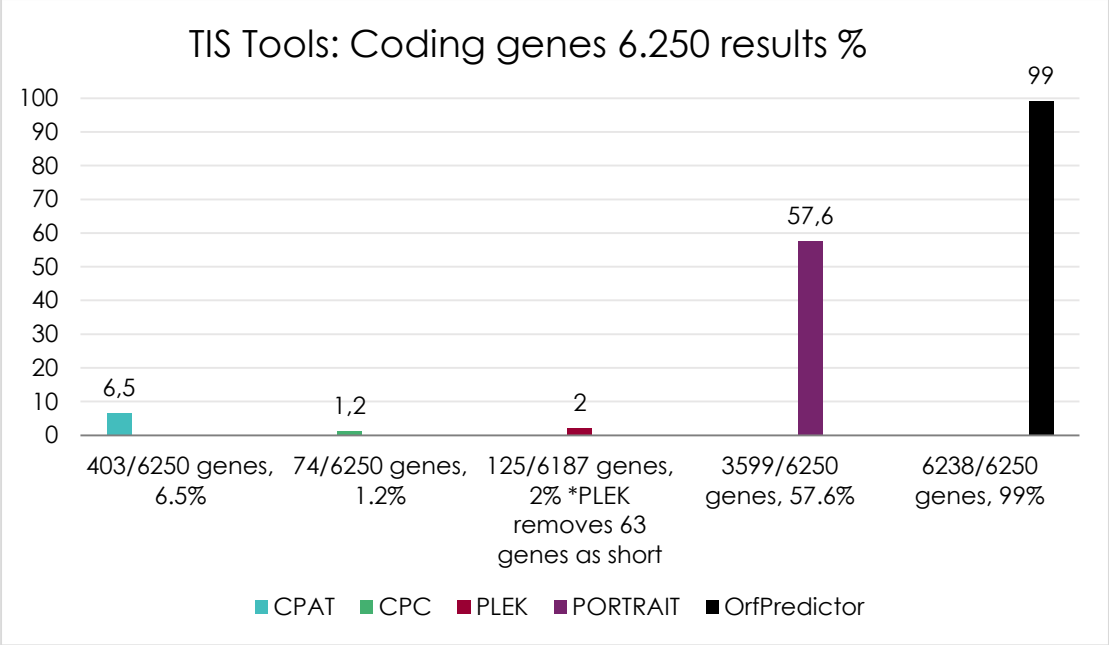


Figure appendix 1.16: TIS Tools results coding for dataset "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

All TIS Tools overall results for datasets, "Coding genes 11.955" & "Long non-coding genes 6.250"

The overall results from the TIS Tools are presented in the chart below. As we can see all tools have gathered very good results, except PORTRAIT and OrfPredictor. (see Figure appendix 1.17).

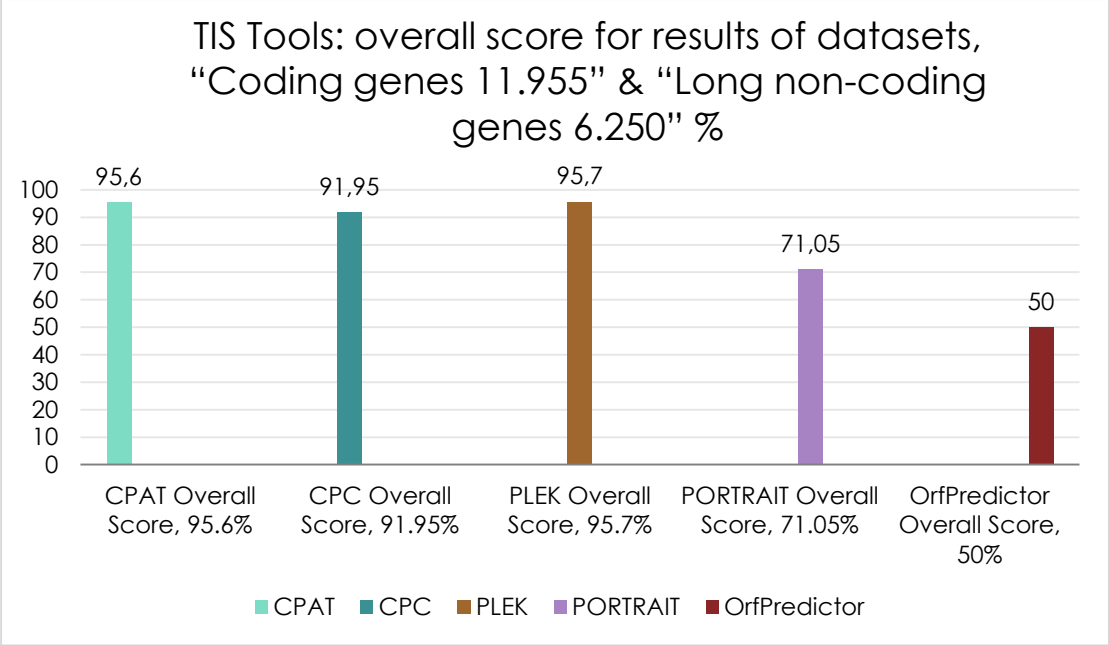


Figure appendix 1.17: Overall results for TIS Tools for two datasets "Coding genes 11.955" & "Long non-coding genes 6.250". This figure has been designed for the purpose of this thesis.

Appendix H D-TIS algorithm description

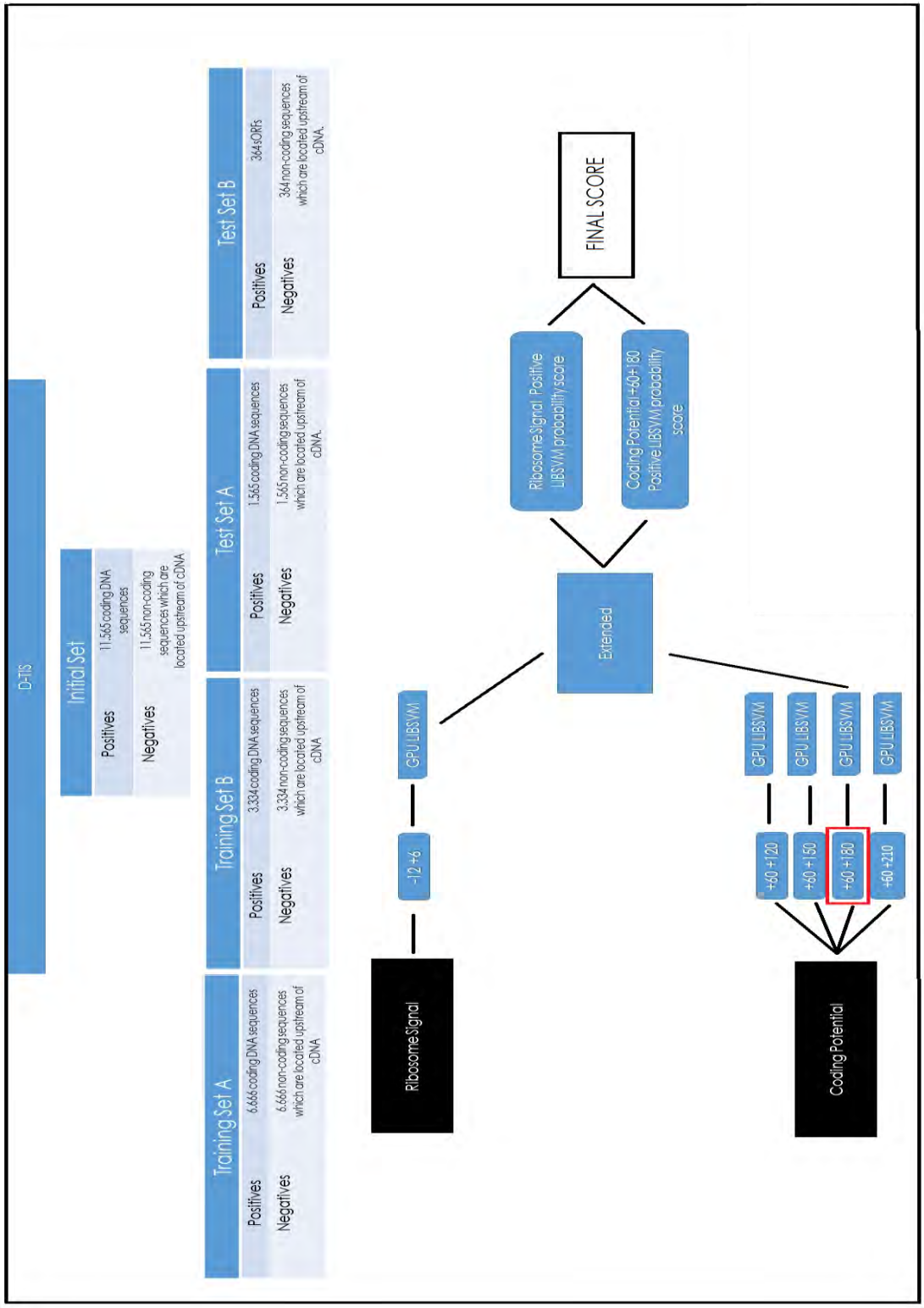


Figure appendix 1.18: How the D-TIS works. This figure has been designed for the purpose of this thesis.