UNIVERSITY OF TRENTO
Department of Computer Science and Information Engineering

International ICT Doctoral School

PhD Dissertation

Machine Learning for Investigating Post-Transcriptional Regulation of Gene Expression

Gianluca CORRADO

Advisors

Andrea PASSERINI
University of Trento

Gabriella VIERO
National Research Council

May 2017

Abstract

RNA binding proteins (RBPs) and non-coding RNAs (ncRNAs) are key actors in post-transcriptional gene regulation. By being able to bind messenger RNA (mRNA) they modulate many regulatory processes. In the last years, the increasing interest in this level of regulation favored the development of many NGS-based experimental techniques to detect RNA-protein interactions, and the consequent release of a considerable amount of interaction data on a growing number of eukaryotic RBPs.

Despite the continuous advances in the experimental procedures, these techniques are still far from fully uncovering, on their own, the global RNA-protein interaction system. For instance, the available interaction data still covers a small fraction (less than 10%) of the known human RBPs. Moreover, experimentally determined interactions are often noisy and cell-line dependent. Importantly, obtaining genome-wide experimental evidence of combinatorial interactions of RBPs is still an experimental challenge.

Machine learning approaches are able to learn from the data and generalize the information contained in them. This might give useful insights to help the investigation of the post-transcriptional regulation. In this work, three machine learning contributions are proposed. They aim at addressing the three above-mentioned shortcomings of the experimental techniques, to help researchers unveiling some yet uncharacterized aspects of post-transcriptional gene regulation.

The first contribution is RNAcommender, a tool capable of suggesting RNA targets to unexplored RBPs at a genome-wide level. RNAcommender is a recommender system that propagates the available interaction data, considering biologically relevant aspects of the RNA-protein interactions, such as protein domains and RNA predicted secondary structure.

The second contribution is ProtScan, a tool that models RNA-protein interactions at a single-nucleotide resolution. Learning models from experimentally determined interactions allows to denoise the data and to make predictions of the RBP binding preferences in conditions that are different from those of the experiment.

The third and last contribution is PTRcombiner, a tool that unveils the combinatorial aspects of post-transcriptional gene regulation. It extracts clusters of mRNA co-regulators from the interaction annotations, and it automatically provides a biological analysis that might supply a functional characterization of the set of mRNAs targeted by a cluster of co-regulators, as well as of the binding dynamics of different RBPs belonging to the same cluster.

Keywords: post-transcriptional gene regulation, RNA-protein interactions, recommender systems, kernel machines, matrix factorization.

Acknowledgements

The first acknowledgement goes to my family for all the support they have given me during this PhD.

The biggest thank goes to my advisors: Andrea Passerini and Gabriella Viero. If it was not for Andrea I probably would have not started my PhD. It's been some challenging and amazing years, and surely I would have not make it through in this way if it wasn't for both of you. I also want to thank Toma Tebaldi for all the work we did together and all the support he provided me with.

I want to acknowledge the reviewers Sander Granneman and Samuel Kaski for their useful comments that contributed to improve this thesis.

I also want to thank the Bioinformatics Group of the University of Freiburg for having me as a visiting PhD student. Thanks to Fabrizio Costa for inviting me to join the group and for the useful help he gave me in all research projects of my PhD. Regarding the experience in Germany I cannot avoid to thank CrossFit Freiburg. Moving to a new town might be scary but you guys made me feel like home from the very first day.

Another big thank goes to all the present (and past) members of the Structured Machine Learning Group of the University of Trento (Stefano Teso, Paolo Dragone and Paolo Morettin). It was a blast working together and sharing the office with you and the insanely noisy vents of our machines. Moreover, I want to thank Francesca Belton and Andrea Stenico from the ICT Doctoral School secretariat for all the help they gave me in the past years.

Last but not least, thank you to Erasmus Student Network Trento (ESN Trento). Thanks to this student association I had the opportunity of meeting a lot of great people including the group of people that I'm happy to call my second family.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In this first chapter, I give the motivation of my research work, and introduce the main contributions of this thesis. I also provide a basic explanation of the structure of the manuscript.

1.1 Motivation

Proteins are responsible for the majority of processes taking place in all prokaryotic and eukaryotic cells. Proteins are produced according to the central dogma of molecular biology (Crick *et al.*, 1970), that explains how they are synthesized through gene transcription and translation. The first process (i.e. transcription) copies a portion of DNA into a messenger RNA (mRNA); while the second one (i.e. translation) translates the information carried by the mRNA into functional proteins. Numerous regulatory steps occur to control the amount of proteins expressed in a cell. Albeit transcriptional control has been well studied and characterized, only in the last years, post-transcriptional regulation called for attention. Importantly, the evidence of a widespread uncoupling between transcriptome (the product of transcription) and proteome (the product of translation) supports the presence of a post-transcriptional regulatory mechanism (Vogel *et al.*, 2010; Tebaldi *et al.*, 2012).

In this work I focus on the study of eukaryotic (mainly human) post-transcriptional regulation. At this level of regulation, proteins and non-coding RNAs (ncRNAs) may regulate mRNA metabolism acting as trans-

factors on the mRNA. Among these, RNA binding proteins (RBPs) and micro RNAs (miRNAs), that are able to bind mRNA molecules and modulate several regulatory processes, are the most studied actors of post-transcriptional regulation. In eukaryotes, each mRNA undergoes a series of post-transcriptional steps before being translated into a functional protein. These include mRNA processing (capping, polyadenylation and splicing), transport, storage, translation and degradation. Elucidating the basic mechanisms of post-transcriptional control is fundamental to gain a full understanding of how gene expression is regulated at different levels. Such knowledge is crucial to understand how defects in post-transcriptional regulation can lead to numerous genetic disorders (Modic *et al.*, 2013) and cancer (Farazi *et al.*, 2011).

The understanding of RNA-protein interactions is an essential point for studying post-transcriptional regulation. For this reason, genome-wide experimental techniques have been developed for detecting interactions (Marchese *et al.*, 2016) both *in vitro* (Ray *et al.*, 2009; Lambert *et al.*, 2014) and *in vivo* (Ule *et al.*, 2003; Granneman *et al.*, 2009; Hafner *et al.*, 2010; König *et al.*, 2010; Kudla *et al.*, 2011; Van Nostrand *et al.*, 2016). The coupling of *in vivo* techniques based on crosslinking, such as CLIP (Ule *et al.*, 2003; Hafner *et al.*, 2010; König *et al.*, 2010; Van Nostrand *et al.*, 2016), CRAC (Granneman *et al.*, 2009) and CLASH (Kudla *et al.*, 2011), with next generation sequencing, allowed the identification of RBP-RNA interactions genome-wide. These techniques, by exploiting substitutions and/or deletions in the RNA sequences, allow to precisely pinpoint the interaction sites. Together, all these techniques, enabled the generation of an unprecendented source of information for the study of post-transcriptional gene regulation.

Despite the continuous advances in the experimental procedures, these techniques are still far from fully uncovering, on their own, the global RNA-protein interaction network. In the scope of this thesis, I want to underline three main shortcomings of the data produced by these experimental techniques. First, the available interaction data still covers quite a small fraction of the known RBPs. Considering human RNA-protein interaction data, the RNA interactome is currently available for less than 10% of the known 1542 manually curated collection of RBPs (Gerstberger *et al.*, 2014). This lack of information is not only related to the cost and time of obtaining these data, but also to experimental problems. For example, the unavailability

of reliable antibodies against certain RBPs, or specific chemical properties of the interaction that complicate the crosslinking make obtaining reliable information of RBP-RNA interaction a challenge. Second, experimentally determined interactions are often noisy and cell-line dependent. Even for RBPs with experimentally determined interactomes, the information is still far from being fully accurate (Marchese *et al.*, 2016). Due to the dependency of these techniques on expression levels and cell lines, some interactions might be missed (false negatives). Additionally, cell stress conditions, that in some cases are induced by the experimental procedures themselves, might produce some technical artifacts that are then mistakenly detected (false positives). Third, these techniques individuate the binding sites of a single RBP of interest in each experiment. An exception might be represented by gPAR-CLIP (Baltz *et al.*, 2012) that allowed to determine the mRNA-bound proteome and its global occupancy profile. Anyhow, gPAR-CLIP does not allow to match binding sites with specific RBPs, therefore it does not give precise information usable to understand how multiple RBPs target the same mRNAs. Even if the combinatorial interaction of multiple RBPs with the mRNA has been well hypothesized and in some cases confirmed (Blaxall *et al.*, 2002; Landthaler *et al.*, 2008), obtaining genome-wide experimental evidence of combinatorial interaction of RBPs is still an experimental challenge.

In conclusion, the increasing interest in post-transcriptional regulation of gene expression stimulated the constant release of new experimental data, paving the way for transdisciplinary research and empowering the cooperation between biologists and computer scientists. Moreover, techniques capable of learning from the data, such as machine learning approaches, are able to generalize the information contained in the data and might give useful insights to help the investigation of post-transcriptional regulation.

1.2 Contributions

With the purpose of helping researchers to unveil some yet uncharacterized aspects of the post-transcriptional gene regulation, in this thesis I propose three machine learning contributions aimed at addressing the three above-mentioned shortcomings of CLIP techniques.

RNAcommender was developed with the aim of generating more com-

plete overviews of RNA-protein interactions. This tool helps with the prediction of genuine RNA targets for uncharacterized RBPs. RNAcommender propagates the interaction information (available from experimental data), considering biologically relevant aspects of the RNA-protein interactions, such as the RBP domain composition and the RNA predicted secondary structure.

Even when the interaction information is available, it is often subject to noise and dependent on the specific cell line in which the experiment was performed. For these reasons the second contribution is ProtScan, a tool that accurately models RNA-protein interactions. ProtScan is based on kernel methods and consensus voting, and it exploits the information obtained with CLIP techniques to build generalized models of the binding preference of RBPs. Learning generalized models from experimentally obtained data allows to reduce the noise and to make predictions of the RBP binding preferences in conditions that are different from those used in the specific experiment (e.g. different cell lines with respect to the one used in the experiment). To give an example, Ferrarese *et al.* (2014) investigated the role of the splice factor PTBP1 in differential splicing of the tumor suppressor gene ANXA7 in glioblastoma. Although there was strong biological evidence for PTBP1 directly binding ANXA7, no binding site was found in a publicly available CLIP-seq dataset for PTBP1. Instead, only a generalized *in silico* model trained on publicly available data was capable to generalize the information and predict PTBP1 binding sites which were then experimentally validated to affect ANXA7 splicing regulation.

The third and last contribution aims at unveiling the combinatorial aspects of the post-transcriptional gene regulation. Although CLIP techniques are able to determine all RNA interactors for a given RBP, they do not directly provide any information on the combinatorial interaction of multiple RBPs (e.g. cooperative interaction of two RBPs in binding the same mRNAs). For this reason, a computational tool named PTRcombiner is proposed. It extracts clusters of mRNA co-regulators from interaction data. PTRcombiner employs a pattern set mining technique based on Boolean matrix factorization to extract the clusters of co-regulator RBPs. Additionally, it provides a biological analysis of the extracted clusters that might suggest some aspects of the functional characterization of the set of mRNAs targeted by a cluster of regulators, and of the binding dynamics of different

RBPs that belong to the same cluster.

1.3 Structure of the thesis

In this chapter I introduced the main focus of my research work. The rest of the thesis is organized as follows. Chapter 2 introduces the related topics in biology, i.e. the key concepts in post-transcriptional gene regulation, and the experimental techniques to detect RNA-protein interactions. Chapter 3 describes the machine learning and data mining techniques that are related to the three research contributions of this thesis: recommender systems, kernel machines, and pattern set mining. Chapter 4 illustrates RNAcommender, the recommender system for predicting RNA-protein interactions for uncharacterized RBPs. The work presented in Chapter 4 has already been published in:

> Corrado G., Tebaldi T., Costa F., Frasconi P. and Passerini A. (2016). RNAcommender: genome-wide recommendation of RNA-protein interactions. *Bioinformatics*, **32**(23), pp. 3627-3634

Chapter 5 discusses ProtScan, a tool for modeling RNA-protein interactions from the available experimental data. The work presented in Chapter 5 has been submitted and it is currently under peer review:

> Corrado G., Uhl M., Backofen R., Passerini A. and Costa F. ProtScan: modeling and prediction of RNA-protein interactions. *Bioinformatics*

Chapter 6 presents PTRcombiner, a data mining tool for unveiling the combinatorial aspects of post-transcriptional gene regulation. The work presented in Chapter 6 has already been published in:

> Corrado G., Tebaldi T., Bertamini G., Costa F., Quattrone A., Viero G., and Passerini A. (2014). PTRcombiner: mining combinatorial regulation of gene expression from post-transcriptional interaction maps. *BMC Genomics*, **15**(1)

Finally, in Chapter 7 I discuss in detail the final remarks related to the work presented in this thesis.

1.4 Personal contributions

I am first author of the published paper presented in Chapter 4. I prepared the data used in the work, contributed to the development of the model, implemented the tool, performed the experimental validation, performed

the comparison with related work, and contributed to the writing of the manuscript.

I am first author of the submitted paper presented in Chapter 5. I contributed to the preparation of the data used in the work, contributed to the development of the model, implemented the tool, performed the experimental validation, performed the comparison with related work, and contributed to the writing of the manuscript.

I am co-first author of the published paper presented in Chapter 6. I contributed to the modification of the method, contributed to the implementation the tool, performed the experimental validation, performed the comparison with related work, and contributed to the writing of the manuscript.

Chapter 2

Biological Background

In this chapter I introduce the biological topics related to my research work. First, I present the central dogma of molecular biology, together with the basics of the transcription and translation. Then, I focus on the post-transcriptional controls and the RNA-protein interactions. Finally, I discuss the experimental techniques, based on crosslinking and next generation sequencing, to detect RNA-protein interactions.

2.1 The flux of genetic information

In this section, I first introduce the central dogma of molecular biology that states how the information flows through DNA, RNA and proteins, and then I explain the two main processes through which functional proteins are synthesized, i.e. transcription and translation.

Proteins are extremely important for all living cells. In order to produce functional proteins, the information contained in the protein coding genes of the DNA is copied, through a process named transcription, into messenger RNAs (mRNAs). The transcription process also copies the information contained in other genes into non-coding RNAs (ncRNAs). While ncRNAs are per se functional, mRNA molecules encode functional proteins, which are generated through a process referred to as translation.

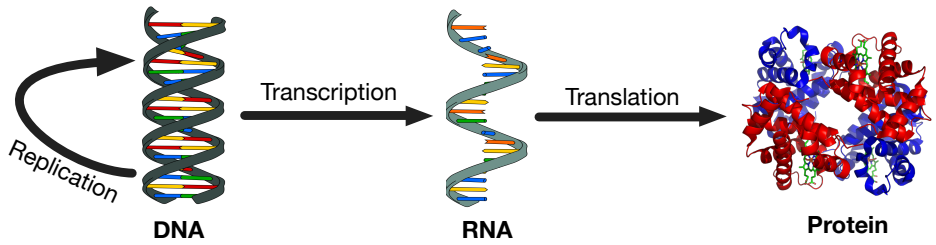


Figure 2.1: Illustration of the flow of information in the central dogma of molecular biology.

2.1.1 The central dogma

The flow of information through DNA, RNA and proteins was first introduced by Crick in 1958 and then refined in Crick *et al.* (1970) in the so called central dogma of molecular biology. The basic version of the central dogma, the one hypothesized in 1958 and shown in Figure 2.1, states that the information contained in the DNA can flow from DNA to DNA, from DNA to RNA, and from RNA to proteins. The first process is named DNA replication and it allows cells to duplicate their entire genome, while the second process, named transcription, makes an RNA copy of sections of DNA that are referred to as genes. Genes encode the information for synthesizing several types of RNA molecules that exert different tasks. Lastly, the information contained in mRNAs flows to proteins, through a process named translation.

Here, I describe transcription and translation that are processes subjected to transcriptional and post-transcriptional regulation of gene expression.

2.1.2 Transcription

In eukaryotes, transcription occurs within the nucleus, where DNA is packaged into nucleosomes and high order chromatin structures. It consists of three stages: initiation, elongation, and termination.

RNA polymerases are the enzymes that drive transcription. In eukaryotic cells, three types of RNA polymerases are present, each with distinct roles and properties. RNA polymerase I (Pol I, Pol A) is responsible for the transcription of all large ribosomal RNAs (rRNAs). RNA polymerase II (Pol II, Pol B), located in a specialized compartment of the nucleus called the

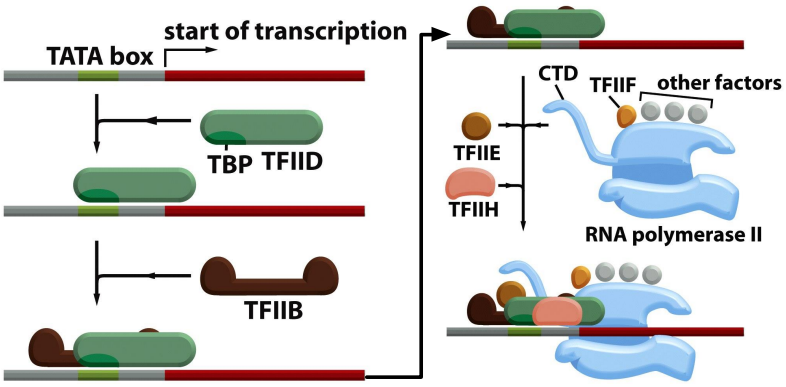


Figure 2.2: RNA polymerase II transcription preinitiation complex (Alberts *et al.*, 2002). The transcription factor II D (TFIID) complex binds, though the TATA binding protein (TBP), to the TATA box in the core promoter of the gene. Then, the transcription factor II B (TFIIB) binds to stabilize the complex. TBIIB also recruits RNA polymerase II and other transcription factors (TFIIE, TFIIF) that help to stabilize the complex. TFIIH promotes the creation of the transcription bubble.

nucleolus, catalyzes the transcription of all messenger RNAs (mRNAs), and ncRNAs such as micro RNAs (miRNAs), small nuclear RNAs (snRNAs), small nucleolar RNAs (snoRNAs) and small interfering RNAs (siRNAs). Finally, RNA polymerase III (Pol III, Pol B), located in the nucleus and in the nucleolus, is in charge of transcribing transfer RNAs (tRNAs), and other small non-coding RNAs (including the small 5S rRNA).

Transcription initiation requires an RNA polymerase and a set of multiple general transcription factors to form a transcription preinitiation complex (Figure 2.2). General transcription factors are a group of proteins involved in transcription initiation and regulation. DNA contains promoter regions that are extremely important for the transcription initiation. Promoter regions can be highly conserved (core promoters) and therefore promote the initiation of transcription for many genes (e.g. TATA box), or located outside core promoter regions. Enhancers and silencers bind transcriptional activators or repressors to increase or decrease transcription. An additional type of these cis-acting elements are the insulators that blocks the interaction between enhancers and promoters to inhibit their subsequent interactions. After the RNA polymerase and the transcription factors have bound the DNA, the newly formed complex opens the two DNA strands and

positions the template strand in the active site of the RNA polymerase.

After that, the transcription enters in its elongation phase. At this step, RNA polymerases acquire enzymes, named elongation factors, that catalyze the unwinding of the DNA double strand and the scanning of the template strand by the RNA polymerases. For every DNA base pair separated by the progressing RNA polymerase, one hybrid DNA-RNA base pair is instantly formed. Then, the two DNA strands rejoin at the end of the transcription bubble while the single-stranded RNA emerges alone. Elongating RNA polymerase II is also associated with a set of factors (such as P-TEFb, SPT5 and TAF-SF1) required for mRNA processing, capping, splicing, and polyadenylation. The 5'-end of the mRNA is capped as soon as it emerges from the exit channel of the polymerase. Then intronic sequences, that do not carry information for assembling proteins, are removed by splicing. Finally, mRNA is cleaveged and then polyadenylation adds a poly(A) tail to its 3'-end.

The last stage is transcription termination, where the complete RNA transcript dissociates and the RNA polymerase is released from the DNA template strand. The termination process varies for each of the three types of RNA polymerases. Pol I and Pol II undergo a factor-dependent termination, where specific transcription termination factors associate with the RNA polymerase to dissociate it from the DNA template strand. Pol I transcribes large rRNAs, when it reads through termination sites the rRNA is cleaved by enzymes. Pol II is associated with the transcription of mRNAs. For Pol II, CPSF (cleavage and polyadenylation specificity factor) and CSTF (cleavage stimulation factor) recruit other proteins to carry out RNA cleavage and polyadenylation. Differently, Pol III terminates the transcription without the involvement of additional factors, because it directly recognizes the termination signal in the sequence of the template strand.

During transcription several levels of control act both locally, to turn on or off individual genes in response to specific needs of the cell, and globally, to maintain the gene expression pattern that shapes cell identity (epigenetic regulation). Transcription initiation is, in particular, the primary level of transcriptional regulation, because targeting the initial step is more energy efficient for the cell. Transcription initiation is regulated by cis-acting elements (enhancers, silencers, insulators) present in the regulatory regions of the DNA, and sequence-specific trans-acting factors that act as activators or

repressors. Still, gene transcription can also be regulated after the initiation phase by targeting the elongation of the RNA polymerases. Also transcription termination can be interpreted as a level of control, because the factors associated with transcription termination indirectly determine the rate of re-initiation.

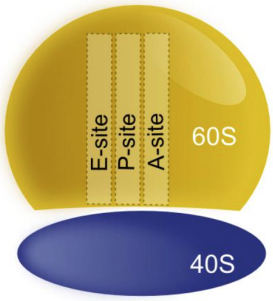
2.1.3 Translation of mRNAs

After transcription and mRNA processing, mRNAs are exported from the nucleus into the cytoplasm, where they can be translated, by ribosomes, into functional proteins. The general structure of mature mRNA (processed) in the cytoplasm is shown in Figure 2.3a. At the two extremities there are the 5' cap (red) that was added during capping, and the poly(A) tail (grey) that was added during polyadenylation. The coding sequence (CDS) (green) contains the actual information needed to assemble the proteins. The nucleotide chain of the CDS determines the amino acid composition of a protein. The code is read in blocks of 3 nucleotides, called codons, and each codon specifies the amino acid that needs to be added on the growing polypeptide chain. Finally, there are two untranslated regions (UTRs) that are sections of the mRNA, at the extremities of the CDS, that are not translated, namely 5' UTR (yellow) and 3' UTR (pink). Their role is mainly associated to regulation processes.

If the main actors of transcription are the transcription factors and the RNA polymerases, in translation the key role is played by ribosomes and transfer RNAs (tRNAs). A ribosome is a large complex composed of ribosomal RNAs (rRNAs) and ribosomal proteins. Figure 2.3b shows a cartoon of an eukaryotic ribosome (also known as 80S ribosome). Eukaryotic ribosomes are composed of two unequal subunits, named small subunit (40S) and large subunit (60S), that assemble to form an 80S ribosome. Ribosomes contain three active sites, named E-, P-, and A-site, where mRNA and tRNAs are located during mRNA translation. tRNAs have a distinctive folded structure with three hairpin loops (Figure 2.3c), one of which contains a sequence called the anticodon. Anticodons match their complementary codons on the mRNA. Each tRNA has its corresponding amino acid (that corresponds to the one encoded by the matched codon) attached to its end.

Translation can be divided in four main steps: initiation, elongation, termination and recycling. Differently from transcription (that occurs in

(a) mRNA structure.



(b) Ribosomal subunits and ribosomal active sites.

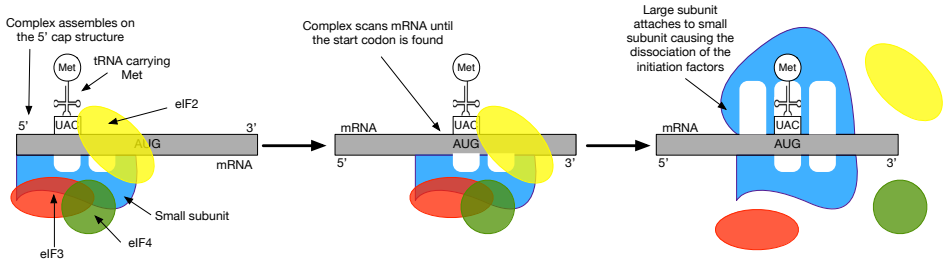
(c) tRNA structure.

Figure 2.3: Structures of the mRNA, the ribosome, and the tRNA.

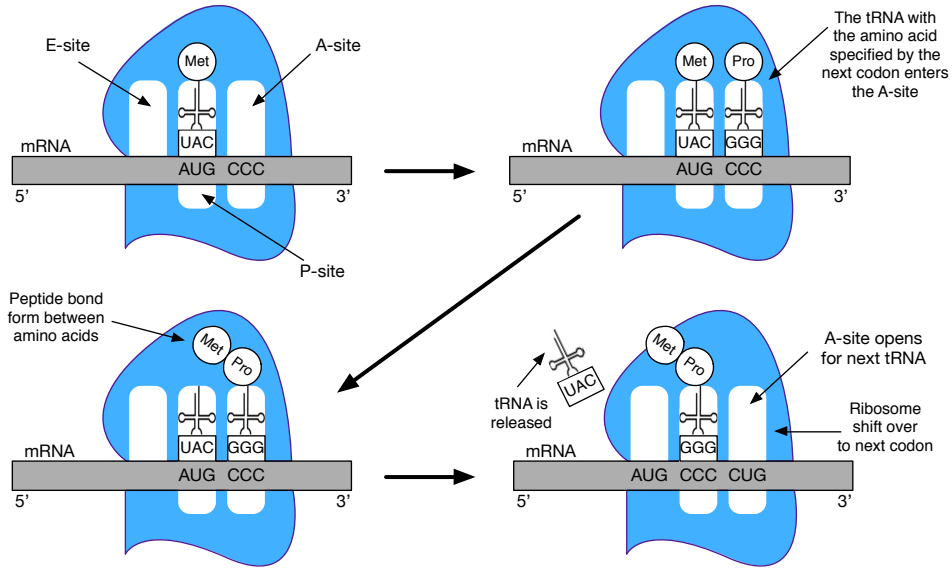
the nucleus), translation takes place in the cytoplasm.

Transcription initiation starts with the translation initiation factor eIF4, that is a large protein complex composed of multiple subunits, binding to the 5' cap of the mRNA. eIF4 also binds, through one of its subunits, to the poly(A)-binding proteins bound to the poly(A) tail of the mRNA, inducing a circularization of the molecule. At the same time, another initiation factor (eIF3) binds to the small ribosomal subunit and loads it on the circularized mRNA at the beginning of the 5' UTR. Then, the complex formed by the small ribosomal subunit and the initiation factors starts scanning the mRNA until it finds a start codon (AUG), that represents the beginning of the CDS. The AUG codon is recognized by a unique tRNA carrying a methionine amino acid that will be removed from the assembled protein. The recognition of the start codon is also catalyzed by the initiation factor eIF2. At this point, the large ribosomal subunit binds to this complex, causing the release of the initiation factors. The 80S ribosome is now assembled around the mRNA (Figure 2.4a).

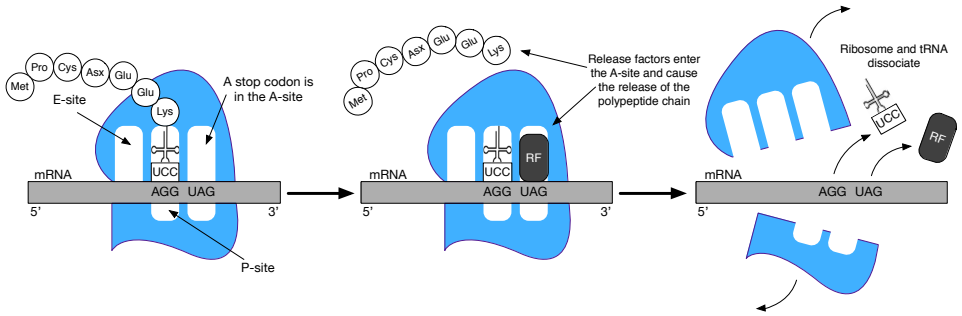
During the elongation phase amino acids are brought together and joined to form a polypeptide chain. This process is directed by elongation factors and it can be divided in three steps, summarized in Figure 2.4b, that are



(a) Translation initiation.



(b) Translation elongation.



(c) Translation termination.

Figure 2.4: Steps of the translation of mRNA.

repeated until a stop codon (that marks the end of the CDS) is encountered. First, a tRNA enters the A-site of the ribosome. This tRNA has the complementary anticodon to the codon in the A-site. Now, two tRNA molecules are side by side in the P- and A-sites of the ribosome, and their amino acids are next to each other. Second, rRNA of the ribosome catalyzes the bond formation between the two adjacent amino acids. The amino acid carried by the tRNA in the P-site is attached to the amino acid of the tRNA in the A-site, and the growing protein chain is temporarily held by the tRNA in the A-site. Third and last, the ribosome and the mRNA slide relative to each other. The tRNA that was in the P-site is shifted into the E-site, the tRNA that was in the A-site is transferred into the P-site. This situates a new codon in the A-site and the growing polypeptide chain in the P-site. The tRNA in the E-site exits the ribosome, and then the steps of elongation may repeat.

Translation terminates when one of the three stop codons (UAA, UAG and UGA) enters the A-site of the ribosome. The stop codons are not recognized by any tRNA, but by release factors. When, the release factors enter the ribosome, they catalyze: the breaking of the bond between the growing polypeptide chain and the tRNA that holds it, the release of the polypeptide chain from the ribosome, and the dissociation of the ribosomes subunits that are now free to associate again and translate another mRNA or the same mRNA another time (ribosome reciclying) (Figure 2.4c).

2.2 Post-transcriptional gene regulation

In many mammals it is possible to observe a profound uncoupling between transcriptome (the product of transcription) and proteome (the product of translation) (Tebaldi *et al.*, 2012), suggesting a widespread presence of gene expression controls also at a post-transcriptional level (Vogel *et al.*, 2010). For this reason, while transcriptional control has been well studied and characterized, a new level of control of gene expression, named post-transcriptional regulation, called for attention. Here RNA-binding proteins and ncRNAs (mostly miRNAs) bind mRNAs to regulate their translation and/or degradation. Furthermore, the involvement of aberrant RBPs, or the synthesis of malfunctioning proteins due to failures in the post-transcriptional regulation steps, often cohorts with the development of

diseases (Glisovic *et al.*, 2008).

2.2.1 Splicing

The pre-mRNA splicing reaction is a fundamental step in the regulation of eukaryotic gene expression. Almost all mammalian genes produce multiple mRNA alternative isoforms through alterations in the choice of splice sites. Pre-mRNA contains exons and introns that are delineated by the 5' splice site at the beginning of an intron and the 3' splice site at its end. Alternative splicing involves changes in the choice of the splice sites by the splicing machinery with the help of splicing factors (as RBPs). During splicing, introns are excised and exons are ligated. The process is catalyzed by a large ribonucleoprotein (RNP) complex called spliceosome, that assembles onto splice sites (or splice junctions). Although several processes alter spliceosome assembly and affect the splice site choice, the best understood alterations in splicing are defined by RBPs that bind to the pre-mRNA and boost or inhibit the spliceosome assembly. Although each regulatory protein can affect many different RNA targets, each transcript is usually targeted by multiple regulators (Vuong *et al.*, 2016).

Even small alterations of the relative spliceosome assembly rates can largely influence the choice of the splicing pattern in a transcript. Through the combinatorial assembly of multiple alternatively spliced exons, genes can produce tens of mRNA isoforms. These isoforms allow to produce proteins of different structures and functions, or to affect mRNA localization, translation or degradation. For example, the broad class of the heterogeneous nuclear ribonucleoproteins (hnRNPs) is strongly associated to the regulation of the splicing machinery. Defective hnRNPs or alterations in their expression level has been associated to a plethora of diseased cellular states, including amyotrophic lateral sclerosis (ALS), Alzheimer's disease and cancer (Geuens *et al.*, 2016). Alternative splicing also plays a critical role in both neuronal development and the function of mature neurons. For this reason, the misregulation of splicing is implicated in multiple neurological disorders (Vuong *et al.*, 2016).

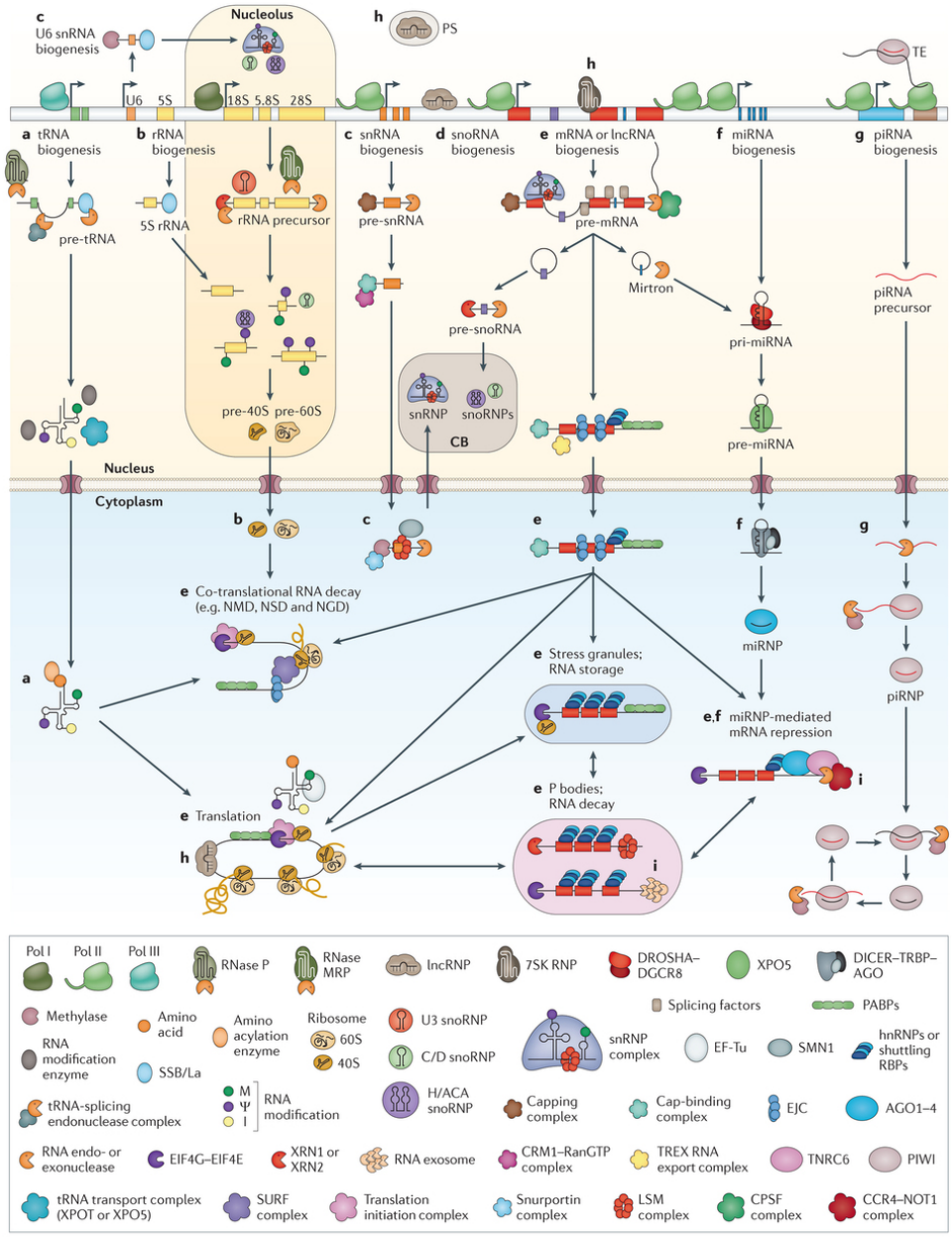


Figure 2.5: Overview of the post-transcriptional gene regulation pathways in eukaryotes (Gerstberger *et al.*, 2014).

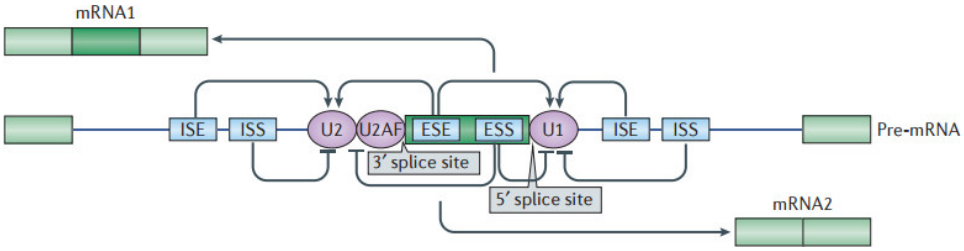


Figure 2.6: Regulation of an alternative exon by RNA-binding proteins (Vuong *et al.*, 2016). Trans-acting RNA-binding proteins (RBPs) interact with cis-sequence elements in the precursor mRNA to facilitate or inhibit the assembly of the spliceosomal machinery at nearby splice sites. The 5' splice site is initially bound by U1 small nuclear ribonucleoprotein (snRNP). The U2 snRNP recognizes the branchpoint and is recruited by the U2AF proteins that are bound between the branchpoint and the 3' splice site. Binding of U1 and U2 allows recognition of an exon in a process called exon definition. An alternative splicing event frequently involves multiple competing weak splice sites that are subject to dynamic regulation by neighbouring cis-elements. These cis-elements include intronic and exonic splicing enhancers (ISE and ESE) and intronic and exonic splicing silencers (ISS and ESS) that recruit activator or repressor RBPs, respectively. These RBPs collectively influence splice site recognition or splice site pairing within the spliceosome. The levels and activity of these trans-acting RBPs control the choice of splice sites for many different transcripts. Activator RBPs binding to enhancer elements are shown as arrows, and repressors binding to silencer elements are shown as inhibitory arrows. Constitutive flanking exons are shown in light green and the alternative exon is shown in dark green.

2.2.2 Polyadenylation

During nuclear mRNA processing, all mRNAs acquire a poly(A) tail of approximately 250–300 adenosine residues in length. Although the addition of a poly(A) tail seems to occur by default, the successive control of its length is highly regulated both in the nucleus and in the cytoplasm, being responsible for the regulation of the stability, transport and translation of mature transcripts.

Poly(A) tails of cytoplasmic mRNA act in synergy with the 5' cap to aid the translation initiation through the stabilization of the closed loop formed by the translation initiation factor eIF4, providing a general (non–mRNA-specific) way of translational regulation (Craig *et al.*, 1998). On the other hand, mRNA-specific translational control is determined by cis-acting regu-

latory sequences, that are mainly present in the 5' and 3' UTRs. These motifs form mRNA-specific ribonucleoprotein complexes (mRNPs), including microribonucleoprotein particles (miRNPs), mRNPs for deadenylation and cytoplasmic polyadenylation complexes, that dynamically vary the length of the poly(A) tail. The length of the poly(A) tail is strongly related to the degree of mRNA translation (Beilharz and Preiss, 2007). The linear view of poly(A) tail length regulation, in which all the mRNAs are polyadenylated during mRNA processing in the nucleus and subsequently deadenylated as the first step towards degradation, indicates only some of the regulatory functions that involve poly(A) tails.

A much more dynamic view of poly(A) tail acquisition, shortening and lengthening better explains the role of the poly(A) tail in the regulation of gene expression. Even though nuclear polyadenylation is a default process, the position at which the 3' UTR of mRNA is cleaved and polyadenylated is highly regulated for numerous transcripts (alternative polyadenylation) (Tian and Manley, 2017). The choice of the cleavage point determines the regulatory signals that will be present in the 3' UTRs of mature transcripts. During stabilization of the translationally silent transcripts, these regulatory signals present in the 3' UTR will mediate mRNA deadenylation by forming deadenylation mRNPs. Translationally inactive mRNPs may accumulate in the cytoplasm, to be quickly reactivated by cytoplasmic poly(A) tail elongation when their encoded proteins are needed (Di Giammartino *et al.*, 2011).

2.2.3 Export

Before mRNAs can be translated into proteins they must be processed to become mature transcripts, and then be exported from the nucleus to the cytoplasm, by crossing through the nuclear pore complexes (NPCs). This process is mediated by transport factors such as the conserved nuclear RNA export factor 1 (NXF1) and its cofactor p15 that, together, bind and export mature mRNAs. Transport through a NPC is accomplished by surmounting the permeability barrier that is created by nuclear pore proteins called FG-nucleoporins. In addition to the export factor and its cofactor, mRNA export involves two complexes that recognize mRNAs while they are still being transcribed: transcription-export complex (TREX) and TREX-2. After transcription and processing, cargo mRNAs from both TREX and

TREX-2 are transferred to NXF1–p15, that directly interacts with the FG-nucleoporins and mediates the transit through the NPC.

Although it is possible that the majority of mRNAs are exported through bulk export pathways, the selectivity of mRNA export has been recently shown (Wickramasinghe and Laskey, 2015). Diverse biological processes, including gene expression (Wickramasinghe *et al.*, 2014), can be regulated by selective mRNA export and, in the majority of these cases, the selectivity is mediated by components of the TREX and TREX-2 complexes. Moreover, there is growing evidence that malfunctions of the mRNA export may contribute to the development of cancer (Culjkovic-Kraljacic and Borden, 2013).

Wickramasinghe and Laskey (2015) hypothesized that the mRNA export selectivity may be linked to the coordinate activity of the production of functionally related proteins by mRNP complexes in post-transcriptional RNA regulons. Functionally related genes that are preferentially transcribed in certain cell states may be (post-transcriptionally) regulated by specific RBPs that recognize sequence elements that are conserved among the mRNAs (Keene, 2007).

2.2.4 Storage and degradation

mRNA decay can be divided into two broad classes. The first represents the mechanisms of quality control that eliminate the production of potentially toxic proteins, while the second includes the mechanisms that lengthen or shorten mRNA half-life for the purpose of changing the abundance of functional proteins. The cytoplasmic decay machinery consists of different types of ribonucleolytic activities, that are combinatorially used depending on the mRNA substrate and cellular conditions. These activities mediate decapping, 5'-to-3' exonucleolytic decay, deadenylation, 3'-to-5' exonucleolytic decay or endonucleolytic cleavage (Schoenberg and Maquat, 2012).

Proteins and ncRNAs associated with mRNAs can influence the rate of mRNA decay in two ways. Directly, by promoting or precluding decay factor binding, and indirectly by influencing the cellular location and/or translational status of the mRNA. For example, by recruiting deadenylases onto target mRNAs through TNRC6A–C proteins, miRNAs can promote mRNA destabilization (Rehwinkel *et al.*, 2005).

Cell state and environmental conditions (e.g. stress conditions) require

rapid adaptations of gene expression. For this reason, RBPs can promote the formation of membrane-less organelles, such as stress granules and processing-bodies (P-bodies). Stress granules and P-bodies are associated with mRNA storage and degradation, respectively, and they are produced in response to different types of environmental conditions (Giménez-Barcons and Díez, 2011).

Imprecise assembly or disassembly of stress-granules and P-bodies can threaten cell stability. In diseased states, mutated RBPs contained in such assemblies are associated with elevate structural disorder, and by consequence with high risk misfolding and formation of toxic protein aggregates, especially in neurons. Many motor-neuron diseases are connected to the accumulation of disordered RBPs, e.g. TDP43 in amyotrophic lateral sclerosis (ALS), or Ataxia 1 in Ataxia (Bossy-Wetzel *et al.*, 2004).

2.2.5 Translation

Although the ribosome has always been perceived as a remarkable molecular machine for reading and translating the genetic code of mRNAs, recent studies have discovered significant functional specificity of many core ribosomal proteins and unveiled greater gene regulatory potential by the ribosome (Xue and Barna, 2012). Accordingly, heterogeneity in the composition of the ribosome provides a platform for extensive diversity in ribosome activity and/or function, paving the way for a level of translational control played by the ribosomal core proteins. For example, it has been shown that a single core ribosomal protein, RPL38, indirectly helps to establish the mammalian body plan by selectively facilitating the translation of subsets of *Hox* mRNAs, genes critically required for formation of the body plan (Xue *et al.*, 2015). In addition to ribosomal core proteins, several additional proteins (associated with ribosomal activity) have been identified as exerting a specific type of regulation of translation. One notorious example is the fragile-X mental retardation protein (FMRP), that that represses mRNA translation by directly binding to the ribosome (Chen *et al.*, 2014). Moreover, a lack of FMRP is associated with the Fragile X syndrome (FXS) (Verkerk *et al.*, 1991; Richter *et al.*, 2015).

ncRNAs are also involved in translational regulation. In fact, many studies on different organisms and conducted with different methods have suggested that miRNAs inhibit the initiation step of translation (Braun

et al., 2012; Fabian and Sonenberg, 2012). This repression of the translation initiation step is also supported by more recent genome-wide analyses of endogenous miRNA targets (Eichhorn *et al.*, 2014).

Diverse arrays of cis-regulatory elements (mRNA sequence motifs or structural elements) have been described to control translation in a specific and coordinated fashion. These regulatory elements, embedded in mRNAs (often in the 5' and 3' UTRs), act as critical regulatory platforms. These hidden RNA regulons may interface with innumerable RBPs to perform translational regulation. As revealed by the studies of RPL38-mediated translation of Hox mRNAs, cis-regulatory elements such as the TIE and IRES-like elements are just beginning to be characterized (Xue *et al.*, 2015). Moreover, 5' UTRs encode different cis-regulatory elements, such as AUGs (uAUGs) (Calvo *et al.*, 2009), upstream open reading frames (uORFs) (Wethmar *et al.*, 2010), and IRES (Arcondéguy *et al.*, 2013). These are cis-acting regulatory elements (sequence motifs or structural elements) that are usually located in the UTRs.

2.2.6 Cooperation and competition of trans-acting factors

I presented many types of post-transcriptional controls, with relevant examples from the literature. From these examples, it is clear that many regulatory tasks are exerted by RBPs and ncRNAs. Cis-regulatory elements interact with trans-acting factors (RBPs and miRNAs) to mediate post-transcriptional regulation processes. A simple mechanism (and still not well understood) to express more regulatory paths is to deploy, at the same time, multiple trans-acting factors in a combinatorial way.

Since the publication of the idea of RNA regulons (Keene, 2007), that shows how multiple mRNAs are co-regulated by one or more sequence-specific RBPs, some examples of multiple trans-acting factor activity have been identified. This multiple interactions can either be cooperative or competitive. Cooperative interactions involve protein-protein interactions that can attach the RNA bound by one RBP to another ribonucleoprotein (e.g. spliceosome assembly). Another type of cooperative interactions happens when two RBPs sandwich the RNA forming a protein-RNA-protein complex. This type of cooperative interaction has been observed in the context of large macromolecules like the exon junction complex (Hennig and Sattler, 2015). Competitive and cooperative trans-factor activity has also been

identified, involving both RBPs and miRNAs (Gerstberger *et al.*, 2014). For example, ELAVL1 competes with miRNAs in regulating a large set of mRNA targets (Kim *et al.*, 2009), while PUM proteins cooperate with miR-221 and/or miR-222 to destabilize the CDKN1B mRNA (Galgano *et al.*, 2008; Jiang *et al.*, 2013)

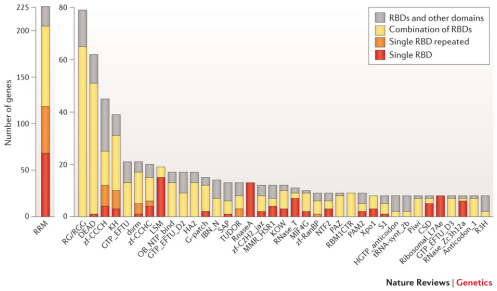
2.3 RNA-protein interactions

As pointed out in Section 2.2, after transcription, mRNA is subjected to several processes that actively contribute to the fate of the mRNA transcript, i.e. polyadenylation, splicing, export, translation, and stability. All these processes are mediated by RNA binding proteins (RBPs) and non-coding RNAs, that, by interacting with the mRNA, promote or suppress post-transcriptional steps that lead to the degradation of mRNAs, repress translation, transport or boost the synthesis of proteins. Moreover, malfunctions in these processes are often associated with diseases. For these reasons, unraveling RNA-protein interactions should provide a solid foundation for understanding post-transcriptional gene regulation, and sketching targeted solutions to treat many diseases.

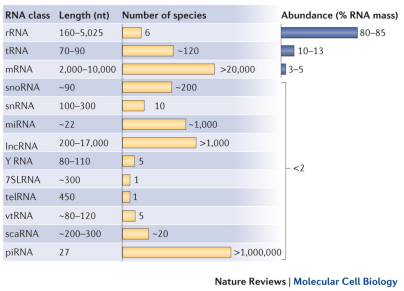
2.3.1 Complexity of RNA-protein interactions

Recently, a census of 1,542 human RBPs has been identified by Gerstberger *et al.* (2014). Some of them are ubiquitously expressed, while others are expressed exclusively in particular tissues. Many RBPs contain canonical RNA-binding domains (RBDs). Each protein might contain multiple repeats of the same RBD or combinations of different RBDs. The most frequently occurring RBD is the RNA recognition motif (RRM). While other well characterized RBDs are KH, DEAD, zinc-fingers, dsrm and many others (Figure 2.7a). However, RNA binding activity is not restricted to proteins containing RBDs. In fact, considerable association with RNA activity have been imputed to several metabolic enzymes lacking canonical RBDs (Baltz *et al.*, 2012; Castello *et al.*, 2012). For this reason, the number of proteins that interact with RNA will likely to grow in the near future.

In eukaryotic cells, RNAs outnumber RBPs. According to Ensembl (Aken *et al.*, 2016) the human genome encondes more than 20,000 different protein



(a) Presence of frequent RBDs in human genes (Gerstberger *et al.*, 2014).



(b) Major classes of eukaryotic RNAs (Jankowsky and Harris, 2015).

Figure 2.7: RNA binding domains in RBPs and types of RNA molecules.

coding genes. Moreover, alternative splicing and other post-transcriptional modifications of RNAs increase the diversity of mRNA.

Different types of RNA are drastically altered in concentration. Usually, rRNAs cover around 80-85% of the cellular RNA mass, followed by tRNAs with 10-13% and by mRNAs with 3-5%, leaving to the other types of RNA less than the 2% of the total RNA mass (Figure 2.7b).

RNAs can be bound by multiple RBPs at the same time. Proteins can bind simultaneously, subsequently, consecutively or in a mutually exclusive fashion. At the same time, most proteins are able to bind multiple RNAs. Considering this scenario and the collection of proteins and RNAs expressed in living cells, the number of possible RNA-protein interactions is utterly large. Moreover, RNAs can also interact with each other, e.g. miRNAs interact with mRNAs to regulate translational efficiency.

RNA-protein interactions can be considered as a massive set of interdependent interactions. Each RNA-protein interaction is governed by inherent affinity for the RNA site, concentration of the protein and the RNA, the competition among other proteins to bind the RNA, and the competition from other RNAs to associate with the protein. Although, the interplay of many RBPs can completely alter the RNA-binding patterns. For this reason, target selection for an RBP rarely complies to simple rules.

The modeling of RNA-protein interaction is an ambitious goal that will deeply help the understanding of post-transcriptional gene regulation, and a critical step towards this goal is the quantitative determination of RNA-protein interactions by experimental techniques.

2.3.2 Methods to study RNA-protein interactions

RNA-protein interactions play a key role in post-transcriptional regulation of gene expression. Therefore, determining such interactions represents an essential step in the investigation of the complex regulatory system of the gene expression. Experimental techniques to study RNA-protein interactions can be categorized in two broad classes: low-throughput and high-throughput techniques. Low-throughput techniques allow to test interactions between a single RBP (or a specific domain of the RBP) and a single transcript (or part of it). For example: X-ray analysis of co-crystallized RNA-protein complexes (Ke and Doudna, 2004), electrophoretic mobility shift assay (EMSA) (Hellman and Fried, 2007), and nuclear magnetic resonance spectroscopy (NMR) (Dominguez *et al.*, 2011). High-throughput techniques, e.g. UV crosslinking and immunoprecipitation (CLIP) (Ule *et al.*, 2003), allow the individuation, with a single experiment, of the genome-wide RNA interactome of an RBP. Often, low-throughput techniques are used to further validate some of the interactions resulted from high-throughput approaches.

In this work I mainly focus on high-throughput techniques for RNA-protein interaction detection. These techniques can be further divided into two classes, i.e. *in vitro* and *in vivo*. The former screens synthetic RNAs 9-10 nucleotide long in a controlled environment. The latter involves complex RNA molecules that are present in living cells. Testing interactions *in vitro* produces affinity profiles for a protein, or an RNA binding domain, towards some artificial short RNA fragments, while *in vivo* RNA-protein interactions are known to be substantially affected by competitive or co-operative interactions that involve other proteins. These protein-protein interactions might significantly alter the binding affinity obtained from *in vitro* experiments (Hennig and Sattler, 2015; Marchese *et al.*, 2016).

2.3.2.1 High-throughput *in vitro* methods

A famous *in vitro* technique is systematic evolution of ligands by exponential enrichment (SELEX) (Ellington and Szostak, 1990). In SELEX an RBP or a single RNA binding domain is evaluated against a library of fixed-length, single-stranded RNAs with largely random sequences. Unbound RNAs are removed from the pool by washing, while the bound ones are amplified by

PCR. This process is iterated until convergence, i.e. the set of washed RNAs is null or, at least, negligible. One possible pitfall of SELEX is the production, during amplification, of PCR artifacts. RNAcompete (Ray *et al.*, 2009) and Bind-n-seq (Lambert *et al.*, 2014) extend SELEX by substituting PCR amplification with microarray analysis and next generation sequencing, respectively. Another method, HiTS-RAP (Tome *et al.*, 2014) also permits the quantitative determination of association and dissociation constants.

2.3.2.2 High-throughput *in vivo* methods

RNA immunoprecipitation (RIP) (Tenenbaum *et al.*, 2000) purifies RNAs associated to a protein in living cells by employing protein-specific antibodies and detecting the target RNAs by microarray (RIP-chip) or sequencing (RIP-seq). One limitation of RIP is the inability to precisely locate the coordinates of the nucleotides that are interacting with the protein. Moreover, the mRNAs identified as putative target of the RBP of interest can be associated to other proteins that, through protein-protein interaction, are bound to the protein of interest. In this case the mRNAs are not direct target of the RBP of interest. In addition, a postlysis reassortments of RNA-protein interaction is possible as demonstrated by the fact that co-immunoprecipitation does not always recapitulate the in vivo state of ribonucleoprotein complexes. This artifact was found for the association of HuR with its target mRNA c-fos that largely result from reassociation of molecules subsequent to cell lysis (Mili and Steitz, 2004).

The introduction of crosslinking and digestion methods, allowed the identification of regions of the RNA that are protected from nuclease digestion by the protein. This new family of experimental techniques takes the name of CLIP (crosslinking and immunoprecipitation) (Ule *et al.*, 2003). Several CLIP variants have been released in the past years. First, HITS-CLIP (Licatalosi *et al.*, 2008; Chi *et al.*, 2009) applied high-throughput sequencing to individuate the RNA residues of the crosslinked fragments. PAR-CLIP (Hafner *et al.*, 2010) boosted the crosslinking efficiency by introducing ribonucleoside analogs in the sample, while iCLIP (König *et al.*, 2010) allowed the individuation of crosslinking sites at a nucleotide resolution. Finally, eCLIP (Van Nostrand *et al.*, 2016) reduced the presence of PCR duplicates in the sequenced RNAs, lowering the false discovery rate of interacting RNA fragments. Also CRAC (Granneman *et al.*, 2009) and

CLASH (Kudla *et al.*, 2011) employ UV crosslinking, but instead of using immunoprecipitation they adopt affinity purification of tagged proteins to increase the specificity of the results. CRAC and CLASH also provided the first demonstration that deletions can be used to map the crosslinking site.

2.4 CLIP techniques

As mentioned in Section 2.3.2, many experimental techniques for detecting RNA-protein interactions have been developed. In this work, I focus on high-throughput approaces based on next generation sequencing. Crosslinking and immunoprecipitation (CLIP), CRAC and CLASH, coupled with high-throughput sequencing allow the genome wide discovery of RNA-protein interactions. The breakthrough of these techniques is the ability of localizing the RNA residues interacting with the protein.

2.4.1 CLIP variants

The first CLIP approach was presented in Ule *et al.* (2003) in combination with high-throughput sequencing. After the success of HITS-CLIP (Licatalosi *et al.*, 2008; Chi *et al.*, 2009), many variants of the method have been developed. PAR-CLIP (Hafner *et al.*, 2010) introduces ribonucleoside analogs, usually 4-thiouridine (4SU), to boost crosslinking efficiency. The addition of 4SU results in thymine (T) to cytosine (C) conversions during the retro-transcription step, indicating the presence of a protein binding site. iCLIP (König *et al.*, 2010) is a protocol designed to obtain information on protein binding sites at a single nucleotide resolution. While in HITS-CLIP the reverse transcriptase is expected to read past the crosslinking site, in iCLIP the amino acids crosslinked to the RNA are expected to work as road block, which frequently results in termination of reverse transcription. These stop sites provide valuable information regarding the crosslinked region, and allow a more accurate localization of the crosslinking site. eCLIP (Van Nostrand *et al.*, 2016) is a novel, faster and more accurate CLIP technique, that significantly reduces the presence of PCR duplicates in the sequenced reads. eCLIP introduces some modifications of the iCLIP protocol to improve the library preparation of RNA fragments. For example: separate ligation of two adapter sequences instead or RNA circularization which results in much higher RNA fragment recovery, and the

inclusion of a size-matched input control (SMInput) which enables efficient background normalization. irCLIP (Zarnegar *et al.*, 2016) adopts the RNA circularization of iCLIP, but instead it employs the TGIRT enzyme in the retrotranscription to improve the efficiency of the reaction. CRAC (Granneman *et al.*, 2009) and CLASH (Kudla *et al.*, 2011), adopt tandem affinity purification of tagged proteins instead of immunoprecipitation. Moreover, CLASH also adopts intermolecular RNA-RNA ligation. The key steps and the main differences among these methods are summarized in Figure 2.8.

All CLIP variants, CRAC and CLASH terminate with high-throughput sequencing (Figure 2.8). These protocols produce a cDNA library of the crosslinked RNAs linked to 5' and 3' adapters, i.e. uninformative RNA sequences that are attached to the begin and end of the crosslinked RNA due to protocol specific reasons. These sequence-specific adapters are first ligated to the RNA before the retrotranscription is performed. Then a sequence specific primer is hybridized to the 3' adapter followed by the addition of the reverse transcriptase.

These techniques overcome the main disadvantages of RIP by introducing three major improvements with respect to the RIP protocol. First, they apply UV irradiation to create covalent bonds between the protein and the RNA, allowing stringent purification and, consequently, an increased signal-to-noise ratio. Second, RNase is used to digest the unbound parts of RNA preserving only the protein binding sites. Last, an accurate purification (multiple washings, SDS-PAGE and blotting) allowed the elimination of non-specific proteins for the antibody and of not-crosslinked RNAs, decreasing the background signal.

Despite the improvements introduced by these techniques, they are still far from being 100% accurate. A large bottleneck is represented by the low RNA output efficiency. For example, the crosslinking efficiency of HITS-CLIP is approximatively between 1 and 5%. Although, PAR-CLIP introduces 4SU to improve crosslinking efficiency, its performance varies among different RBPs. Moreover, 4SU is toxic to most organisms at the concentrations used for the PAR-CLIP experiments (Kemény-Beke *et al.*, 2006). The problem of toxicity, induced by the experimental procedures themselves, might produce some technical artifacts that are then mistakenly detected. In some recent work, the use of extremely short 4tU labeling allowed the individuation of the genome-wide RNA processing kinetics (Barrass *et al.*,

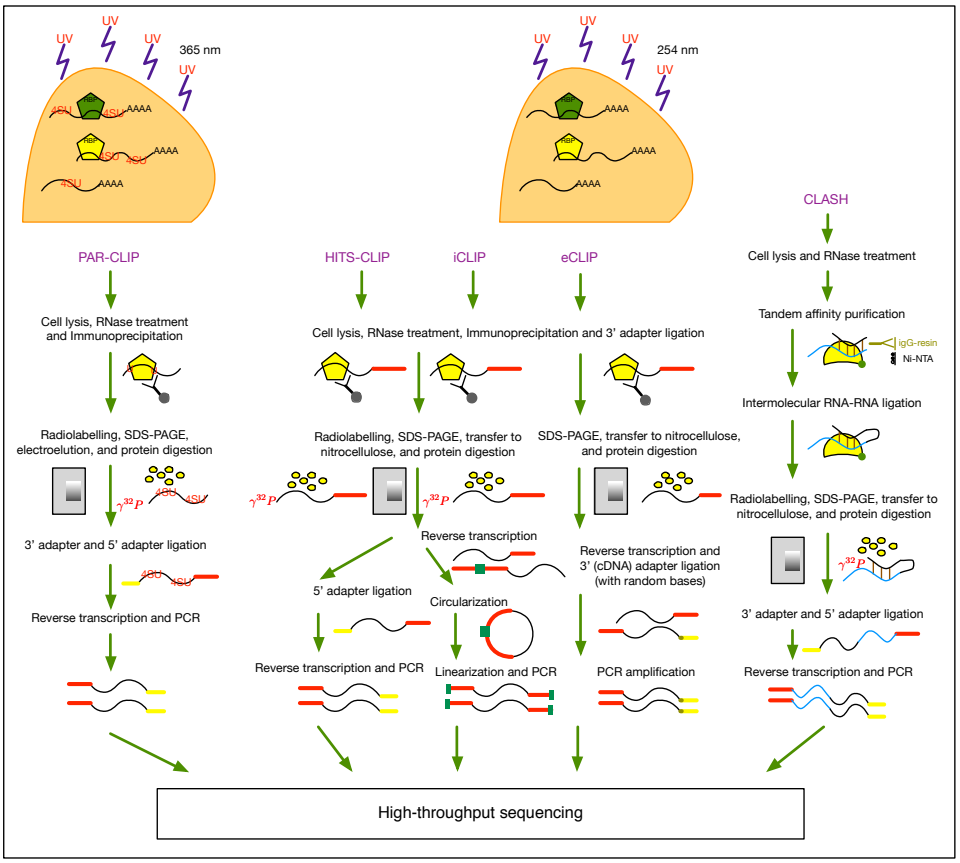


Figure 2.8: Schematic workflow of CLIP and differences of each method. Adapted from Zhang *et al.* (2015) and Van Nostrand *et al.* (2016). First, the cells are irradiated with UV light in order to form covalent bonds between proteins and RNAs. PAR-CLIP uses 4SU and higher UV wavelength to improve crosslinking. Second, the cells are lysed and RNase is added to the lysis buffer to digest unbound RNA, preserving RNA-protein complexes. Third, using the convenient antibody for the protein of interest, the complexes are immunoprecipitated to separate the complexes that involve the protein of interest from all the others. CLASH adopts tandem affinity purification, where the second step (nickel beads) is done under completely denaturing conditions to reduce background. Fourth, the RNAs of the RNA-protein complexes are radiolabeled (except for eCLIP) and the samples are denatured, separated by SDS-PAGE, and blotted in order to visualize the complexes and cut the corresponding bands. Then, the selected bands are incubated with proteinase K, that digests the protein and allows the release of crosslinked RNA fragments. The RNAs are linked to adapters and reverse-transcribed, and, finally, the cDNA is amplified by PCR and sequenced with high-throughput techniques.

2015).

2.4.2 Computational analysis of CLIP data

From the sequenced data to the binding sites of a protein, the data must undergo some steps that are performed *in silico*.

First, the adapter sequences must be removed. One useful library that allows, among other things, the removal of adapters is FASTX-Toolkit[1] from the Hannon Lab. Another tool, specifically developed for trimming adapters from reads sequenced with Illimina technology, is Trimmomatic (Bolger *et al.*, 2014).

Second, the crosslinked RNA fragments are aligned to the reference genome. This step of the analysis pipeline is the most computationally expensive. During the alignment, millions of short (less than 100 nucleotides) RNA sequences require to be aligned to very long genomes, e.g. the human genome is 3 billion nucleotides long, or entire transcriptomes of tens of thousands of mature mRNAs. An efficient alignment tool is Bowtie (Langmead *et al.*, 2009). By using Burrows-Wheeler indices of the genome, it allows extremely fast alignment keeping a small memory footprint. Bowtie 2 (Langmead and Salzberg, 2012) and TopHat[2] also allow to consider splice junction in the alignment of reads. Kallisto (Bray *et al.*, 2016) is an alignment tool based on the idea of probabilistic pseudoalignment for rapidly determining the compatibility of reads with targets, without the need for alignment.

Last, peak extraction is performed to spot the parts of the genome with a significant enriched binding sites. Different tools have been developed for calling peaks in samples obtained from different CLIP variants. PARalyzer (Corcoran *et al.*, 2011) exploits T-C conversions in PAR-CLIP data to identify binding sites, Piranha (Uren *et al.*, 2012) uses reverse transcription stop sites of iCLIP to identify crosslinking regions with a single nucleotide precision, and CLIPper[3] is developed for eCLIP to maximize the accuracy of the called peaks for this novel protocol.

[1]FASTX-Toolkit is available at http://hannonlab.cshl.edu/fastx_toolkit/
[2]TopHat is available at https://ccb.jhu.edu/software/tophat/
[3]CLIPper is available at https://github.com/YeoLab/clipper

2.4.3 Databases

The growing popularity of post-transcriptional gene regulation research is charming more and more research laboratories around the world to produce valuable data. Together with this growth of the data, the retrieval of all the information became less and less straightforward. For this reason, databases that collect the available RNA-protein interaction information started to be released. In some cases, they aggregate highly heterogeneous data, produced by different laboratories, using different organisms, experimental techniques, library preparation procedures, sequencing platforms and analysis pipelines.

doRiNa (a database of RNA interactions in post-transcriptional regulation) (Blin *et al.*, 2015) aggregates interaction information in human, mouse, worm, and fly. It includes both protein-mRNA and miRNA-mRNA interactions. For each experiment it is possible to download a BED file annotating the experimentally determined binding regions.

iCount (Curk *et al.*, 2011) aggregates data obtained from the analysis, with a dedicated pipeline, of iCLIP experiments. It contains interaction information for human RBPs, obtained in different cell lines and tissues.

CLIPdb (Yang *et al.*, 2015) represents the first effort to aggregate data produced with HITS-CLIP, PAR-CLIP and iCLIP, processing the sequenced reads with the same analysis pipeline. The database includes results of experiments performed in different cell lines of multiple organism (human, mouse, worm and yeast). In its newest version, named POSTAR (Hu *et al.*, 2016), it also includes information regarding RNA secondary structures, disease-associated variants, gene expression and function.

AURA (Atlas of UTR Regulatory Activity) (Dassi *et al.*, 2014) incorporates, among other things, the interactions of RBPs and miRNAs with UTRs, which are the untranslated regions of the mRNA, for both human and mouse. Its light version contains one single text file that annotates all the interactions, from all experimental techniques and for both human and mouse.

ENCODE was born in 2004 as the encyclopedia of DNA elements (Consortium *et al.*, 2004). With the increasing interest of post-transcriptional controls, it started to also include experiments addressing RNA-protein interactions (Sloan *et al.*, 2016). To date, ENCODE contains human interaction information, regarding hundreds of RBPs, obtained in the same laboratory, with the same technique (eCLIP), and analyzed using the same

pipeline. Moreover, for each RBP, the published results involve two technical replicates, and sometimes also different cell lines.

The release of CLIP data, broadly encouraged transdisciplinary research, empowering cooperation between biologists and computer scientists. Computational techniques can prove valuable tools for the analysis of the newly released data. Moreover, techniques capable of learning from the data, such as machine learning approaches, are able to generalize the information contained in the data and might give useful insights to help the investigation of post-transcriptional regulation.

Chapter 3

Machine Learning Background

In this chapter I introduce the machine learning topics related to my research work. First, I present recommender systems, then I describe kernel methods, and last I illustrate pattern set mining.

Some of the topics, i.e. string kernels, Neighborhood Subgraph Pairwise Distance Kernel, Boolean matrix factorization, are described in a more formal fashion. The main reason is that these contributions are used as is in my research work. Other topics (e.g. recommender systems) are discussed in a less formal way. The idea is to give an overview of the research topics to provide the reader with the tools to better understand the models developed in my research work and presented in the next chapters.

3.1 Recommender systems

The concepts introduced in this section are extracted from Shapira *et al.* (2011) and Aggarwal (2016).

Recommender systems are techniques devoted at providing suggestions of useful items to a user. The suggestions provided by a recommender system (or recommendations) aim at assisting the users in various decision-making processes. Some notable examples are: what items to buy, what music to listen to, or what news to read. Recommender systems are important assets

when dealing with the information overload online users are subjected to. Nowadays, recommender systems serve as one of the most dominant information discovery tools on the web. Several research efforts have been spent in developing such systems, and in the past 10 years many recommendation techniques have also been successfully deployed in commercial environments.

The recommendation problem can be described as producing an educated guess, based on the available information, of the response of a user to new items, suggesting items that are unknown to the user for which the predicted response is high. User-item responses (or ratings) can be numerical values (e.g., 1–5 stars), ordinal values (e.g., strongly agree, agree, neutral, disagree, strongly disagree), or binary values (e.g., like/dislike or interested/not interested). Moreover, ratings can be obtained explicitly, for example through ratings/reviews submitted by users in the system, or implicitly, for example from the purchase history.

The rest of the section is organized as follows: first I introduce the main classes of recommender systems, and then I explore in a more detailed fashion such classes that include neighborhood-based collaborative filtering techniques, model-based collaborative filtering techniques, content-based recommender systems, and hybrid recommender systems.

3.1.1 Classes of recommendation techniques

Recommendation approaches, that are commonly used in a plethora of applications, aim at suggesting personalized recommendations for each user. Personalized approaches can be divided in content-based and collaborative filtering methods, as well as hybrid techniques that blend these two types of methods.

Content-based approaches (Balabanović and Shoham, 1997; Billsus and Pazzani, 2000) identify the common aspects of items that have been positively rated by a user, and then recommend to the user new items that share these aspects. Recommender systems based on content usually suffer from two problems: limited content analysis and over-specialization (Shardanand and Maes, 1995). Limited content analysis arises when there is scarce information on the users or the content of the items. For example, in some cases, the accurate content of items may be challenging to obtain for some classes of items (e.g. music or images), or in other cases the content of an item is insufficient to determine its quality. Differently, over-specialization

comes as a side effect of the approach used in content-based systems, where high predicted ratings for some items are issued when these items are similar to the ones liked by the user. Over-specialization happens when only highly interrelated items, that are often already obvious to the users, are recommended.

Instead of relying on content information, collaborative filtering approaches exploit rating information of other users and items in the system. The underlying idea is that the rating of a target user for a new item is likely to be analogous to the one of another user, if both users have rated other items in a similar way. Similarly, the target user is likely to rate two items in a comparable fashion, if other users have given similar ratings to these two items. Collaborative filtering techniques surmount some of the limitations of content-based approaches. For instance, items that suffer from limited content information can still be recommended through the feedback of other users. Moreover, in collaborative filtering, the quality of items is evaluated by peer users, instead of relying on content that may be a bad indicator of quality. Finally, collaborative filtering approaches can recommend items with very different content, provided that other users have already demonstrated interest for these different items.

Collaborative filtering techniques can be arranged in two general classes named neighborhood- and model-based methods. In neighborhood-based collaborative filtering (Adomavicius and Tuzhilin, 2005), the available user-item ratings are directly used to infer ratings for new items. This can be done in a user-based or item-based fashion. User-based systems (Shardanand and Maes, 1995; Konstan *et al.*, 1997) evaluate the interest of a target user for an item using the ratings for this item by users that have similar rating patterns (neighbors). On the other hand, item-based approaches (Linden *et al.*, 2003; Deshpande and Karypis, 2004) predict the rating of a user for an item based on the ratings of the user for similar items, where the similarity of two items is defined by the amount of users that have rated these items in a similar way. Differently from neighborhood-based systems, that perform the prediction directly from the stored ratings, model-based approaches (Takács *et al.*, 2008, 2009) use the ratings to learn a predictive model. Important aspects of both users and items are captured by a set of model parameters, that are learned from the available ratings and later used to predict new user-item responses.

Finally, hybrid recommendation approaches combine characteristics of multiple recommendation techniques. For example, hybridized models usually achieve better performance than content-based or collaborative filtering approaches. The combination can happen in various ways, for instance, by aggregating their individual predictions into a single more robust one, or by introducing content information into a collaborative filtering model. Several studies have demonstrated that hybrid recommendation approaches provide more accurate recommendations than pure content-based or collaborative methods, especially when few ratings are available (Adomavicius and Tuzhilin, 2005).

3.1.2 Neighborhood-based collaborative filtering

Neighborhood-based methods rely on either user-user or item-item similarity to make recommendations from the available ratings. The concept of neighbor requires the determination of either similar users or similar items.

3.1.2.1 User-based neighborhood models

I start by discussing the user-based method, where user-based neighborhoods are defined in order to spot users in the system that are similar to the target user for whom the rating predictions want to be computed. In this setting, the similarity function ($sim(u, v)$) is based on the previous ratings specified by the users, taking into account user specific biases such as different scales of ratings, or item interaction discrepancies.

Let R be the $n \times m$ rating matrix where r_{ui} represents the rating given by the user u to the item i. Then I_u denotes the set of items rated by the user u, and given users u and v, $I_u \cap I_v$ represents the set of items rated by both users. Usually rating matrices are sparse, implying in most of the cases that $I_u \cap I_v = \varnothing$.

A simple similarity function between two users u and v can be computed by the cosine function of the raw ratings of the users defined by

$$sim(u, v) = RawCosine(u, v) = \frac{\sum_{i \in I_u \cap I_v} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_u \cap I_v} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} r_{vi}^2}} \quad (3.1)$$

However, different users might be biased towards liking most items, whereas other users might be biased towards not liking most of the items

(Breese *et al.*, 1998). In order to address this bias, the user-specific mean rating μ_u is defined. It evaluates the average rating given by a user to all the items he/she has rated as

$$\mu_u = \frac{\sum_{i \in I_u} r_{ui}}{|I_u|} \quad \forall u \in \{1, \ldots, n\} \tag{3.2}$$

Then, by introducing mean ratings into Equation 3.1, the Pearson correlation coefficient between users u and v can be defined as

$$sim(u,v) = Pearson(u,v) = \frac{\sum_{i \in I_u \cap I_v} (r_{ui} - \mu_u) \cdot (r_{vi} - \mu_v)}{\sqrt{\sum_{i \in I_u \cap I_v} (r_{ui} - \mu_u)^2} \cdot \sqrt{\sum_{i \in I_u \cap I_v} (r_{vi} - \mu_v)^2}} \tag{3.3}$$

The traditional definition of $Pearson(u,v)$ requires μ_u and μ_v to be computed only over the items in $I_u \cap I_v$. However, it is reasonably common (and computationally less expensive) to compute each μ_u just once for each user u, according to Equation 3.2.

One way of defining the neighbors of the target user would be to select the k users with the highest similarity to the target one. However, since the most similar k users might not have rated a specific item of interest for the target user, the closest k users are selected separately for each predicted item, such that each of these k users have specified a rating for that item. Let $P_u(i)$ be the set of the k nearest users to target user u, who have specified a rating for item i. The weighted average of these ratings can be used to predict the rating for that item. Again, there is the problem that different users may provide ratings on different scales, and therefore mean-centered ratings are used. They are computed by

$$s_{ui} = r_{ui} - \mu_u \quad \forall u \in \{1, \ldots, n\} \tag{3.4}$$

obtaining a overall neighborhood-based prediction function defined as

$$\hat{r}_{ui} = \mu_u + \frac{\sum_{v \in P_u(i)} sim(u,v) \cdot s_{vi}}{\sum_{v \in P_u(i)} |sim(u,v)|} \tag{3.5}$$

One variant of the above mentioned prediction function, includes the use of the Z-score z_{ui} instead of the mean-centered ratings s_{ui} (Howe and Forbes, 2008). The Z-score further divides s_{ui} by the standard deviation σ_u

of the observed ratings of the user u:

$$\sigma_u = \sqrt{\frac{\sum_{i \in I_u} (r_{ui} - \mu_u)^2}{|I_u| - 1}} \quad \forall u \in \{1, \ldots, n\} \tag{3.6}$$

then the standardized rating is computed as

$$z_{ui} = \frac{r_{ui} - \mu_u}{\sigma_u} = \frac{s_{ui}}{\sigma_u} \tag{3.7}$$

obtaining a prediction function defined as

$$\hat{r}_{ui} = \mu_u + \sigma_u \frac{\sum_{v \in P_u(i)} sim(u, v) \cdot z_{vi}}{\sum_{v \in P_u(i)} |sim(u, v)|} \tag{3.8}$$

One problem with the Z-score is that the predicted ratings might frequently lie outside the range of the admissible ratings. Nevertheless, they can still be used to rank the items in order of desirability for a particular user.

Another modification that can be brought inside the prediction function, involves the so called amplified similarity, where

$$sim(u, v) = Pearson(u, v)^{\alpha} \tag{3.9}$$

with $\alpha > 1$ it is possible to amplify the importance of the similarity in the weighting of Equation 3.5 and 3.8.

3.1.2.2 Item-based neighborhood models

In item-based models, neighborhoods are constructed in terms of items rather than users. Therefore, similarities are computed between items. This time, before computing the item similarities the rows of the rating matrix R are centered to a zero mean. Similarly, to the user-based case, the ratings are mean-centered with respect to μ_i by computing

$$s_{ui} = r_{ui} - \mu_i \quad \forall i \in \{1, \ldots, m\} \tag{3.10}$$

Let U_i be the set of users that have rated the item i. Given items i and j, $U_i \cap U_j$ represents the set of users that have rated both items. Then, the

Pearson correlation between the items i and j is defined as follows

$$sim(i,j) = Pearson(i,j) = \frac{\sum_{u \in U_i \cap U_j} s_{ui} \cdot s_{uj}}{\sqrt{\sum_{u \in U_i \cap U_j} s_{ui}^2} \cdot \sqrt{\sum_{u \in U_i \cap U_j} s_{uj}^2}} \tag{3.11}$$

Considering the case in which there is interest in determining the rating of the target item i of a user u. First, the k-nearest neighbors to item i need to be computed according to Pearson correlation. Let the k-nearest neighbors of items i, for which the user u has specified ratings, be denoted by $Q_i(u)$. The predicted value is then defined as the weighted average value of the raw ratings by

$$\hat{r}_{ui} = \frac{\sum_{j \in Q_i(u)} sim(i,j) \cdot r_{uj}}{\sum_{j \in Q_i(u)} |sim(i,j)|} \tag{3.12}$$

The basic idea of item-based predictions is to leverage the ratings obtained by the same user on similar items. For example, considering a movie recommendation system, the neighboring items will typically be movies of a similar genre, and the rating history of the same user on such movies is a very reliable predictor of the interests of that user.

3.1.2.3 Strengths and Weaknesses

Neighborhood methods are simple and intuitive approaches, and therefore have several advantages. First, they are easy to implement and debug and it is often easy to justify why a specific item is recommended (especially in item-based methods). The main disadvantage of these methods is that the offline phase is impractical in large-scale settings. For example, the user-based method requires at least $O(n^2)$ to compute the pairwise similarity between users, and this is computationally expensive when dealing with tens of millions of users. Another disadvantage of these methods is their poor resilience to sparsity. When the number of mutually rated items between two users is small, it induces unreliable similarity values.

3.1.3 Model-based collaborative filtering

Model-based methods try to summarize the information contained in the rating data by employing machine learning techniques. Therefore, the model building phase (or training) is performed prior to the prediction phase.

Model-based recommender systems have three main advantages with respect to neighborhood-based methods. First, the space efficiency, because the number of parameters of the learned model is much lower than the number of entries in the rating matrix. Second, model-based systems do not require the preprocessing step of neighborhood-based models, that is quadratic in either the number of users or items. Third and last, the summarization process of model-based approaches is less prone to overfitting.

A plethora of machine learning approaches have been successfully applied to numerous classification and regression tasks, that represent special cases of the collaborative filtering (or matrix completion) task. Anyhow, machine learning models for classification and regression can be generalized to the matrix completion task (Billsus and Pazzani, 1998). In the classification (or regression) problem, there is a definite separation between feature and class variables and between training and test data, while in the matrix completion problem, these distinctions do not exist. It is not straightforward to directly generalize data classification models to the collaborative filtering problem, especially when the great majority of the ratings are missing. For example, collaborative filtering models, such as latent factor models, demonstrated effective in solving matrix completion, but they are not considered competitive models in the context of data classification.

The flourishing interest in collaborative filtering led to the generalization of many classification and regression techniques to the scope of matrix completion. This list of machine learning approaches goes from naive Bayes (Miyahara and Pazzani, 2000) to neural networks (Salakhutdinov *et al.*, 2007). However, here I focus on latent factor models, and especially matrix factorization models, because they are strictly related to my research work.

Latent factor models, such as matrix factorization, leverage dimensionality reduction approaches to fill in the missing entries in the rating matrix. If user-based neighborhood methods leverage user-wise correlations and item-based neighborhood methods leverage item-wise correlations, dimensionality reduction methods, such as matrix factorization, exploit both the user and item correlations present in the rating matrix to create reduced representations of both users and items.

3.1.3.1 Basics on matrix factorization

In the basic matrix factorization model, a $n \times m$ rating matrix R is factorized (with a certain extent of approximation) into a $n \times k$ matrix U and a $m \times k$ matrix V: $R \approx UV^\top$. Each column of U (or V) is named latent vector or latent component, while each row of U (or V) is named latent factor. Each row $\mathbf{u}_i$ of U is called user factor and it is composed of k entries corresponding to the affinity of the user i towards the k concepts used to model the rating matrix. Similarly, each row $\mathbf{v}_j$ of v is called item factor, and it represents the affinity of the item j towards these k concepts.

Each rating r_{ij} of the matrix R can be approximated ($\hat{r}_{ij}$) by the vector product of the user factor of user i and the item factor of item j:

$$r_{ij} \approx \hat{r}_{ij} = \mathbf{u}_i \cdot \mathbf{v}_j = \sum_{s=1}^{k} u_{i_s} \cdot v_{j_s} \tag{3.13}$$

Various matrix factorization methods have been proposed in the past years, and the main differences among them lie in the constraints imposed on U and V (e.g. non-negativity of the latent vectors) and the nature of the objective function (e.g. minimizing the Frobenius norm). These discrepancies define the applicability of the matrix factorization model to different real-world scenarios.

3.1.3.2 Unconstrained matrix factorization

The fundamental matrix factorization case is the unconstrained one, where no constraints are imposed on the factor matrices U and V. The factor matrices U and V should be estimated such that R and $UV^\top$ are as close as possible, and this can be done by defining an optimization problem as follows

$$\min_{U,V} J = \frac{1}{2} \parallel R - UV^\top \parallel^2 = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{m} \left(r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right)^2 \tag{3.14}$$

where $\| \cdot \|^2$ represents the squared Frobenius norm. The smaller the objective function is, the better the approximation of the factorization $R \approx UV^\top$ will be. Gradient descent methods may be applied to optimize such approximations. Let e_{ij} be the approximation error, that is the difference between

the entries of the matrix R and their predicted values defined by

$$e_{ij} = (r_{ij} - \hat{r}_{ij}) = (r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js}) \tag{3.15}$$

Usually, in the context of recommender systems, the rating matrix R contains several missing entries, and therefore, the objective function written in Equation 3.14, and the error function of Equation 3.15 would be undefined. Hence, the optimization problem, that estimates the factor matrices U and V, needs to be rewritten accounting only for the observed entries of R. Note that after the estimation of the optimal U and V the entire rating matrix can be approximated in one shot ($R \approx UV^{\top}$), including the previously missing entries.

Let $S = \{(i, j) : r_{ij} \text{ is observed}\}$ be the set of indeces of observed ratings in the matrix R, then the objective function for incomplete matrices, can be computed only over the observed entries in S turning the optimization problem into

$$\min_{U,V} J = \frac{1}{2} \sum_{(i,j) \in S} e_{ij}^2 = \frac{1}{2} \sum_{(i,j) \in S} \left(r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right)^2 \tag{3.16}$$

In real settings, it is almost always the case that the rating matrix R is extremely sparse, and therefore the number or ratings considered in the model optimization is too low to avoid overfitting. A common approach for addressing this problem is to add regularization terms to the objective function. Regularization diminishes the propensity of the model to overfit the data at the expense of introducing a bias in the model. The idea is to discourage very large values of the entries of the factor matrices U and V in order to promote stability. A regularization term, $\frac{\lambda}{2}(\| U \|^2 + \| V \|^2)$, is added to the objective function, where λ is the regularization parameter.

The regularized objective function is defined as

$$\min_{U,V} J = \frac{1}{2} \sum_{(i,j)\in S} e_{ij}^2 + \frac{\lambda}{2} \sum_{i=1}^{n} \sum_{s=1}^{k} u_{is}^2 + \frac{\lambda}{2} \sum_{j=1}^{m} \sum_{s=1}^{k} v_{js}^2 \tag{3.17}$$

$$= \frac{1}{2} \sum_{(i,j)\in S} \left(r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n} \sum_{s=1}^{k} u_{is}^2 + \frac{\lambda}{2} \sum_{j=1}^{m} \sum_{s=1}^{k} v_{js}^2 \tag{3.18}$$

The model parameters, u_{is} and v_{js}, are learned during the optimization. The optimization can be performed, for example, with gradient descent techniques. One needs to compute the partial derivative of J with respect to the model variables u_{iq} and v_{jq} that are defined as

$$\frac{\partial J}{\partial u_{iq}} = \sum_{j:(i,j)\in S} \left(r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right)(-v_{jq}) + \lambda u_{iq} \quad \forall i, q \tag{3.19}$$

$$= \sum_{j:(i,j)\in S} (e_{ij})(-v_{jq}) + \lambda u_{iq} \quad \forall i \in \{1, \ldots, n\}, q \in \{1, \ldots, k\} \tag{3.20}$$

$$\frac{\partial J}{\partial v_{jq}} = \sum_{i:(i,j)\in S} \left(r_{ij} - \sum_{s=1}^{k} u_{is} \cdot v_{js} \right)(-u_{iq}) + \lambda v_{jq} \quad \forall j, q \tag{3.21}$$

$$= \sum_{i:(i,j)\in S} (e_{ij})(-u_{iq}) + \lambda v_{jq} \quad \forall j \in \{1, \ldots, m\}, q \in \{1, \ldots, k\} \tag{3.22}$$

After computing the derivatives the model parameters are updated as follows

$$u_{iq} = u_{iq} - \alpha \frac{\partial J}{\partial u_{iq}} \quad \forall i \in \{1, \ldots, n\}, q \in \{1, \ldots, k\} \tag{3.23}$$

$$v_{jq} = v_{jq} - \alpha \frac{\partial J}{\partial v_{jq}} \quad \forall j \in \{1, \ldots, m\}, q \in \{1, \ldots, k\} \tag{3.24}$$

This gradient descent procedure represents one way of optimizing the objective function and therefore estimate U and V that better approximate the rating matrix R.

3.1.3.3 Non-negative matrix factorization

Non-negative matrix factorization (NMF) may be employed when rating matrices are non-negative. The main advantage of this approach is not necessarily related to the accuracy of the approximation, but to the level of interpretability of the learned user and item latent factors. Because of the interpretable nature of non-negative decomposition, it is easy to map these aspects to clusters (Zhang *et al.*, 2006).

The key difference from unconstrained matrix factorization is that the model parameters U and V must be non-negative. Therefore, the formulation of the objective function in NMF is stated as follows

$$\begin{array}{ll} \min\limits_{U,V} & J = \frac{1}{2} \parallel R - UV^\top \parallel^2 \\ \text{s.t} & U \geq 0 \\ & V \geq 0 \end{array} \tag{3.25}$$

Although NMF can be used for any non-negative matrix, the interpretability advantages are mostly visible in cases in which the users can only specify their appreciation to items, but in no way they can specify the dislike. Such rating matrices, that represent the so called implicit feedback data, include unary ratings matrices or matrices in which the entries correspond to the activity frequency (that is non-negative).

A helpful aspect of the implicit feedback setting is that it is sometimes possible to set the unspecified entries to zero, instead of treating them as missing values. For this reason, here I address the non-negative matrix factorization problem on a fully specified rating matrix.

As in the case of unconstrained matrix factorization, regularization terms can be added to the objective function to improve the quality of the solution. The basic idea is to add the penalties $\frac{\lambda_1 \|U\|^2}{2} + \frac{\lambda_2 \|V\|^2}{2}$ to the objective function obtaining

$$\begin{array}{ll} \min\limits_{U,V} & J = \frac{1}{2} \parallel R - UV^\top \parallel^2 + \frac{1}{2}\lambda_1 \parallel U \parallel^2 + \frac{1}{2}\lambda_2 \parallel V \parallel^2 \\ \text{s.t} & U \geq 0 \\ & V \geq 0 \end{array} \tag{3.26}$$

where $\lambda_1 > 0$ and $\lambda_2 > 0$ are the regularization parameters, while $\| \cdot \|^2$ is the squared Frobenius norm of the model parameters.

3.1.4 Content-based recommender systems

Content-based recommender systems, are a different class of recommendation techniques that, differently from approaches to collaborative filtering, deal with scenarios in which items can be represented with a descriptive set of attributes. In some cases, when descriptions of the items are available, only the user's ratings on other items are sufficient to discover meaningful recommendations (Balabanović and Shoham, 1997). Content-based recommender systems attempt at matching users to items that are similar to what the users have liked in the past. Differently from collaborative systems, that explicitly exploit the ratings of all the users in the system, content-based recommender systems focus on the target user ratings and the characteristics of the items liked by the user. Therefore, in content-based the contribution of other users to the recommendations issued for the target user is deeply marginal, if not completely absent. Summarizing, content-based systems rely on two types of data sources: a description of the items in terms of content-centric attributes, and a user profile that is generated from the user feedback regarding the items in the system.

Content-based systems are particularly handy when dealing with new items with few available ratings (item cold-start). These types of methods enable to perform the recommendation also in such settings because they leverage the attributes of the new items to make predictions. On the other hand, content-based systems do not use the ratings of other users, and for this reason they still suffer from the user cold-start problem. Furthermore, not exploiting the ratings of other users reduces the diversity and novelty of the recommended items. Often, the recommended items may be obvious for the target user, or even items that the user has already consumed in the past. This is due to the fact that recommendations will always be driven towards items with similar attributes to the ones the target user has consumed before.

Content-based systems are widely applied to scenarios in which a large amount of item information is available. They work with a large variety of item characteristics, usually encoded in unstructured data that must be converted into standardized descriptions (e.g. keywords).

3.1.4.1 Main components of content-based systems

Building and using a content-based systems to perform recommendation implies three steps: the (offline) preprocessing, the (offline) learning, and the (online) prediction. The two offline steps are used to generate a model that is often a classification or regression model. This model is subsequently employed in the online generation of recommendations for the users.

Preprocessing and feature extraction. Content-based systems are employed in a large variety of domains, e.g. web pages, news, music, etc. Usually, the descriptive features are extracted from heterogeneous sources and converted into keyword-based vector-space representations of the items. The proper extraction of the most informative features, that are strongly domain specific, is essential for the effective operation of content-based recommender systems.

Content-based learning of user profiles. Content-based models are specific to a given user. Therefore, by taking into consideration the past history of a target user (user feedback), a (user-specific) model is built to predict item preferences of the given user. User feedbacks are used in conjunction with the attributes of the items in order to assemble the training data, and, subsequently, construct a learning model. This stage is often very similar to standard classification or regression tasks, depending on whether the user feedback is categorical (e.g. binary act of selecting an item), or numerical (e.g. ratings or buying frequency).

Filtering and recommendation. At this step, the most of the job is already done. In fact, the learned model from the previous step is used to recommend items to target users. The only factor to take into account is that recommendations are performed online, and therefore it is important to focus on efficiency because the predictions require to be performed in real time.

3.1.5 Hybrid recommender systems

Hybrid recommender systems combine two or more recommendation techniques (e.g. collaborative filtering with content-based) in the hope of avoiding the limitations of any individual approach and therefore improve the

recommendation performance. Hybrid recommender systems have been successfully applied to many different domains, e.g. music (Tiemann and Pauws, 2007) or movies (Christakou *et al.*, 2007). Here I present simple and common techniques to obtain hybrid recommendation approaches. Anyhow the hybridization may occur in several different ways.

Weighted. In weighted recommender systems the score of a recommended item is computed from the results of all of the available recommendation techniques present in the system. One simple example is the linear combination of recommendation scores. The benefit of a weighted hybrid is that all of the contributions are aggregated in a straightforward way and it is easy to assign to a specific recommender present in the hybrid the credit for a good recommendation or the demerit for a bad one (and eventually adjust the hybrid accordingly).

Switching. A switching hybrid recommender uses some criterion to switch between recommendation techniques. For example, in a content and collaborative hybrid if the content-based system cannot make a recommendation with sufficient confidence (because of a poor description of the item), then a collaborative recommendation is attempted. Alternatively, if the content-based system is suffering of over-specialization for a particular class of items, then the collaborative technique may provide the ability to propose recommendations that are relevant and not close in a semantic way to the items that received a high rating. Since the switching criteria must be determined, switching hybrids introduce additional complexity to the recommendation model.

Mixed. This technique presents together recommendations from more than one technique. The mixed hybrid avoids the item cold-start problem because it is possible to rely on the content-based component to recommend new items on the basis of their descriptions even if they have not been rated by any user. However, this hybridization method does not elude the user cold-start problem, since both content-based and collaborative methods require some data about user preferences to initiate the recommendation process for new users.

Feature combination. A way to merge content-based and collaborative filtering approaches is to use content-based information as additional features data for each example and use collaborative filtering techniques over this refined feature representation. This type of hybrid allows the system to consider collaborative data without relying on it exclusively, reducing the sensitivity of the system to the number of users who have rated an item. Conversely, it enables the system to exploit information about the explicit similarity of items that would be otherwise unintelligible in collaborative systems.

Cascade. In this hybridization approach, one recommendation technique is used to produce a rough ranking of candidates and successively a second technique is used to refine the recommendation of a smaller candidate set of items. Because the second step of cascade targets only those items for which additional discrimination is needed, it is more efficient than, for example, a weighted hybrid that applies all of its techniques to all items. In addition, cascade is noise-tolerant because the ratings given by the first recommendation technique can only be refined, but not overturned.

3.2 Kernel methods

The concepts presented in this section are extracted from Shawe-Taylor and Cristianini (2004).

Solution based on kernel methods are composed by two main modules: a part that computes the mapping into the embedding or the feature space, and a machine learning algorithm able to discover useful linear patterns in that space. These types of approaches, called kernel methods are successful for mainly two reasons. First, they employ efficient algorithms that detect linear relations. Second, in this section I will introduce a computational shortcut that allows to efficiently represent linear patterns in high-dimensional spaces to provide satisfactory representational power.

The rest of the section is organized as follows. First, I present the well-known linear regression model called ridge regression. Second, I explain what kernels are and how they can embed non-linearly separable input features into high-dimensional spaces where a linear separation is more likely to exist. Then I show some mathematical properties of kernels. And finally, I

give examples of kernels on vectorial input data as well as structured inputs such as strings and graphs.

3.2.1 Linear regression

Consider the problem of finding a homogeneous real-valued linear function

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \mathbf{w}^\top \mathbf{x} = \sum_{i=1}^{n} w_i x_i \tag{3.27}$$

that represents the best interpolation of a dataset $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_\ell)\}$ of points $\mathbf{x}_i \in \mathbb{R}^n$ and corresponding labels $y_i \in \mathbb{R}$. g is a linear function, that given the features of $\mathbf{x}_i$, predicts a label as close as possible to the label y_i present in the dataset (for all i's). Namely

$$|y - g(\mathbf{x})| = |y - \mathbf{w}^\top \mathbf{x}| \approx 0 \tag{3.28}$$

In the hypothetical situation where the dataset comes from $(\mathbf{x}, g(\mathbf{x}))$, its cardinality is equal to the number of dimensions ($\ell = n$), and all the points are linearly independent it is possible to find $\mathbf{w}$ by simply solving a system of linear equation $X\mathbf{w} = \mathbf{y}$, where $X \in \ell \times n$ is the matrix representing the dataset and $\mathbf{y}$ is the the column vector representing the labels. In the case the number of points in the dataset is less than the number of dimensions, then there are different $\mathbf{w}$ vectors that exactly describe the data, and usually the one at minimum norm is preferred. Conversely, when the number of points in the dataset exceeds the number of dimensions and there is a noise source in the generation process, then the aim should be finding an approximate solution of the interpolation problem, usually the one yielding minimum error. In general, a mixed situation is occurring. Therefore, by mixing the two strategies the aim is to find $\mathbf{w}$ yielding small norm and error.

Let ξ be the error of the linear function on an example, i.e. $|y - g(\mathbf{x})| = |\xi|$. The aim is to find a function that minimizes these errors. It is usual to account for the sum of the squared errors over the available data, which can be defined by

$$\mathcal{L}(g, S) = \mathcal{L}(\mathbf{w}, S) = \sum_{i=1}^{\ell} \xi^2 = \sum_{i=1}^{\ell} (y_i - g(\mathbf{x}_i))^2 \tag{3.29}$$

This well studied problem is also known as least squares approximation. Let $\xi = \mathbf{y} - X\mathbf{w}$, then Equation 3.29 can be rewritten as

$$\mathcal{L}(\mathbf{w}, S) = \| \xi \|_2^2 = (\mathbf{y} - X\mathbf{w})^\top (\mathbf{y} - X\mathbf{w}) \tag{3.30}$$

and the optimal $\mathbf{w}$ can be found by computing the derivatives of the loss function with respect to the parameters $\mathbf{w}$ and setting them to zero

$$\frac{\partial \mathcal{L}(\mathbf{w}, S)}{\partial \mathbf{w}} = -2X^\top \mathbf{y} + 2X^\top X\mathbf{w} = \mathbf{0} \tag{3.31}$$

obtaining

$$X^\top X\mathbf{w} = X^\top \mathbf{y} \tag{3.32}$$

If the inverse matrix of $X^\top X$ exists the least squares problem can be solved as

$$\mathbf{w} = (X^\top X)^{-1} X^\top \mathbf{y} \tag{3.33}$$

In the majority of the situations the problem is ill-conditioned, meaning that $X^\top X$ is not guaranteed to be invertible. In these cases, it is indicated to search for approximate solutions, by restricting the choice of functions through the, so called, regularization. The simplest regularization approach is represented by seeking for functions with small norm of the $\mathbf{w}$ parameters. Adding regularization to the loss function of Equation 3.29 gives the notorious optimization problem called ridge regression (Hoerl and Kennard, 1970) and defined by

$$\min_{\mathbf{w}} \mathcal{L}_\lambda(\mathbf{w}, S) = \min_{\mathbf{w}} \lambda \| \mathbf{w} \|^2 + \sum_{i=1}^{\ell} (y_i - g(\mathbf{x}_i))^2 \tag{3.34}$$

where λ is a positive real number that controls the trade-off between norm and loss, defining the degree of regularization.

Again the optimization problem can be solved by zeroing the derivatives of the loss function with respect to the parameters of $\mathbf{w}$ obtaining

$$X^\top X\mathbf{w} + \lambda\mathbf{w} = (X^\top X + \lambda I_n)\mathbf{w} = X^\top \mathbf{y} \tag{3.35}$$

where I_n is the $n \times n$ identity matrix. In this case, if $\lambda > 0$ then $(X^\top X + \lambda I_n)$

is always invertible and the solution is given by

$$\mathbf{w} = (X^\top X + \lambda I_n)^{-1} X^\top \mathbf{y} \tag{3.36}$$

Alternatively, Equation 3.36 can be rewritten in terms of $\mathbf{w}$ obtaining

$$\mathbf{w} = \lambda^{-1} X^\top (\mathbf{y} - X\mathbf{w}) = X^\top \alpha \tag{3.37}$$

demonstrating that $\mathbf{w}$ can be expressed as a linear combination of the data points $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i \mathbf{x}_i$, with $\alpha = \lambda^{-1}(\mathbf{y} - X\mathbf{w})$. Therefore:

$$\begin{aligned} \alpha &= \lambda^{-1}(\mathbf{y} - X\mathbf{w}) \\ &\Rightarrow \lambda\alpha = \mathbf{y} - XX^\top \alpha \\ &\Rightarrow (XX^\top + \lambda I_l)\alpha = \mathbf{y} \\ &\Rightarrow \alpha = (G + \lambda I_\ell)^{-1}\mathbf{y} \end{aligned} \tag{3.38}$$

where $G = XX^\top$, and $G_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. The prediction function is now turned into

$$g(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle = \left\langle \sum_{i=1}^{\ell} \alpha_i \mathbf{x}_i, \mathbf{x} \right\rangle = \sum_{i=1}^{\ell} \alpha_i \langle \mathbf{x}_i, \mathbf{x} \rangle = \mathbf{y}^\top (G + \lambda I_\ell)^{-1} \mathbf{k} \tag{3.39}$$

where $k_i = \langle \mathbf{x}_i, \mathbf{x} \rangle$. Now there are two ways for optimizing the ridge regression of Equation 3.34: the primal solution that directly computes the weight vector (Equation 3.36), and the dual solution that expresses the weights as linear combination of the dataset points (Equation 3.38).

In the dual solution, the information contained in the dataset is given by the inner products between pairs of examples, and it is encoded in the, so called, Gram matrix $G = XX^\top$ of size $\ell \times \ell$. In the same way the prediction of a new example just requires to compute the inner products of the new example with the examples in the dataset. When the number of examples (ℓ) is lower than the number of the features (n), the dual formulation allows to compute the $\ell \times \ell$ Gram matrix instead of the $n \times n$ matrix ($X^\top X$) that is part of the primal solution to the problem. As explained in the next section the benefits of the dual formulation of the problem are way more impactful than just improving the efficiency in the case the number of examples is lower than the number of features.

3.2.2 Defining non-linear mappings: kernels

In the majority of the cases, the relations between variables in the dataset are non-linear. Following the overall strategy, now it is time to try to map the input variables into a new feature space where the relations of interest can be expressed in linear form and therefore be detected with a linear model, such as the ridge regression.

Let ϕ be an embedding map defined by

$$\phi : \mathbf{x} \in \mathbb{R}^n \longmapsto \phi(\mathbf{x}) \in \mathbb{R}^N \tag{3.40}$$

where usually $n < N$. The purpose of ϕ is to turn non-linear dependencies into linear ones, casting the dataset into $\hat{S} = \{(\phi(\mathbf{x}_1), y_1), \ldots, (\phi(\mathbf{x}_\ell), y_\ell))\}$, and the problem into looking for relations in the following form

$$|y - g(\mathbf{x})| = |y - \langle \mathbf{w}, \phi(\mathbf{x}) \rangle| = |\xi| \tag{3.41}$$

Even though the primal solution can be applied, it is usually impractical to deal with high-dimensional $N \times N$ matrices. Considering the dual formulation of the problem, only inner products between pairs of data points $\langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$ need to be computed. The predictive function of the dual formulation $g(\mathbf{x}) = \mathbf{y}^\top (G + \lambda I_\ell)^{-1} \mathbf{k}$ uses the Gram matrix $G = XX^\top$ that is composed by entries $G_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle$, and the vector $\mathbf{k}$ is composed by entries $\mathbf{k}_i = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}) \rangle$.

Sometimes, the inner products can be computed as direct function of the input features, avoiding the explicit computation of the mapping ϕ. This shortcut takes the name of kernel trick and it is performed through the so called kernel function (Aizerman *et al.*, 1964).

Definition 3.2.1. (Kernel) A kernel is a function κ that for all $\mathbf{x}, \mathbf{z} \in X$ satisfies

$$\kappa(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle$$

where ϕ is a mapping from X to an inner product feature space F

$$\phi : \mathbf{x} \longmapsto \phi(\mathbf{x}) \in F$$

3.2.3 Valid kernels

Given a set of vectors $S = \{\mathbf{x}_1, \ldots, \mathbf{x}_\ell\}$, the entries of the $\ell \times \ell$ Gram matrix G are represented by $G_{ij} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. When evaluating the inner products in a feature space represented by the feature map ϕ (with a kernel κ) the correspondent Gram matrix contains $G_{ij} = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ (also known as kernel matrix). By definition, Gram matrices are symmetric, which implies that $G_{ij} = G_{ji}$ and that $G = G^\top$. Moreover, they contain all the information needed to compute the pairwise distances within the points in the dataset S.

Definition 3.2.2. (Positive semi-definite matrix) A symmetric matrix $A \in \ell \times \ell$ is positive semi-definite if its eigenvalues are all non-negative, which holds if and only if $\mathbf{v}^\top A \mathbf{v} \geq 0$ for all $\mathbf{v} \in \mathbb{R}^\ell$.

Proposition 3.2.1. Valid kernels are represented by Gram matrices that, for all possible datasets, are positive semi-definite.

Proposition 3.2.2. Closure properties. Let κ_1 and κ_2 be valid kernels over $X \times X$, $X \subseteq \mathbb{R}^n$, $a \in \mathbb{R}^+$, $f(\cdot)$ a real valued function on X, $\phi : X \longrightarrow \mathbb{R}^N$, κ_3 a kernel over $\mathbb{R}^N \times \mathbb{R}^N$ and B a symmetric positive-definite $n \times n$ matrix. Then the following functions are valid kernels:

1. $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z}) + \kappa_2(\mathbf{x}, \mathbf{z})$
2. $\kappa(\mathbf{x}, \mathbf{z}) = a\kappa_1(\mathbf{x}, \mathbf{z})$
3. $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_1(\mathbf{x}, \mathbf{z})\kappa_2(\mathbf{x}, \mathbf{z})$
4. $\kappa(\mathbf{x}, \mathbf{z}) = f(\mathbf{x})f(\mathbf{z})$
5. $\kappa(\mathbf{x}, \mathbf{z}) = \kappa_3(\phi(\mathbf{x}), \phi(\mathbf{z}))$
6. $\kappa(\mathbf{x}, \mathbf{z}) = \mathbf{x}^\top B \mathbf{z}$

3.2.4 Basic Kernels

In this section I provide examples of the polynomial (Boser *et al.*, 1992) and Gaussian kernels (Boser *et al.*, 1992; Wahba *et al.*, 1999). Here, I show how basic kernels are used to compute the similarity between vectorial inputs. As I will show in the next section, kernels are not limited to vectorial inputs, but they can represent the similarity between structured objects like strings and graphs.

3.2.4.1 Polynomial kernel

Definition 3.2.3 (Polynomial kernel)**.** Given a kernel κ_1, the derived polynomial kernel is defines as

$$\kappa(\mathbf{x}, \mathbf{z}) = p(\kappa_1(\mathbf{x}, \mathbf{z}))$$

where $p(\cdot)$ is any polynomial with positive coefficients. Often, it also comes in the following form

$$\kappa_d(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + R)^d$$

defined over a vector space X of n dimensions, where R and d are parameters.

By applying the binomial theorem the polynomial kernel κ_d can be expanded as follows

$$\kappa_d(\mathbf{x}, \mathbf{z}) = \sum_{s=0}^{d} \binom{d}{s} R^{d-s} \langle \mathbf{x}, \mathbf{z} \rangle^s \tag{3.42}$$

The features corresponding to $\kappa_d(\mathbf{x}, \mathbf{z})$ are all the functions $\phi_{\mathbf{i}}(\mathbf{x}) = \mathbf{x}^{\mathbf{i}} = x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}$ where $\mathbf{i} = (i_1, \dots, i_n) \in \mathbb{N}^n$ satisfies $\sum_{j=1}^{n} i_j \leq d$. By induction it is possible to show that the dimension of the feature space associated to a polynomial kernel $\kappa_d(\mathbf{x}, \mathbf{z}) = (\langle \mathbf{x}, \mathbf{z} \rangle + R)^d$ is $\binom{n+d}{d}$.

Note that the parameter R allows to control, to some extent, the relative weightings of the different degree monomials. Then Equation 3.42 can be written as

$$\kappa_d(\mathbf{x}, \mathbf{z}) = \sum_{s=0}^{d} a_s \hat{\kappa_s}(\mathbf{x}, \mathbf{z}) \tag{3.43}$$

where $\hat{\kappa_s}(\mathbf{x}, \mathbf{z})$ is a s degree polynomial kernel and $a_s = \binom{d}{s} R^{d-s}$. Hence, increasing R decreases the relative weighting of the higher order polynomials.

Finally, the polynomial kernel of degree d can be recursively computed using the lower degree polynomial kernels:

$$\kappa_d(\mathbf{x}, \mathbf{z}) = \kappa_{d-1}(\mathbf{x}, \mathbf{z})(\langle \mathbf{x}, \mathbf{z} \rangle + R) \tag{3.44}$$

3.2.4.2 Gaussian kernel

Definition 3.2.4 (Gaussian kernel)**.** The Gaussian kernel is defined, for all $\sigma > 0$, by

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(- \frac{\| \mathbf{x} - \mathbf{z} \|^2}{2\sigma^2} \right)$$

Note that it is not mandatory to employ the Euclidean distance in the input space. For example, considering a kernel $\kappa_1(\mathbf{x}, \mathbf{z})$ with a feature mapping ϕ_1 into a space F_1, it is still possible to create a Gaussian kernel in F_1 by recognizing that

$$\| \phi_1(\mathbf{x}) - \phi_1(\mathbf{z}) \|^2 = \kappa_1(\mathbf{x}, \mathbf{x}) - 2\kappa_1(\mathbf{x}, \mathbf{z}) + \kappa_1(\mathbf{z}, \mathbf{z}) \tag{3.45}$$

and obtaining the following Gaussian kernel

$$\kappa(\mathbf{x}, \mathbf{z}) = \exp\left(- \frac{\kappa_1(\mathbf{x}, \mathbf{x}) - 2\kappa_1(\mathbf{x}, \mathbf{z}) + \kappa_1(\mathbf{z}, \mathbf{z})}{2\sigma^2} \right) \tag{3.46}$$

The parameter σ of the Gaussian kernel plays a role similar to the degree d in the polynomial kernel, i.e. controlling the flexibility of the kernel. Small σ values are similar to large values of d, producing kernel matrices that are similar to the identity matrix. Usually, this configuration allows classifiers to fit any type of labels promoting overfitting and, consequently, yielding poor generalization power. On the other hand, large σ values deliberately reduce the kernel to a constant function, introducing the impossibility to learn any non-trivial classifier.

It is difficult to visually picture the feature space corresponding to a Gaussian kernel. Elements of the feature space can be represented as a function in a Hilbert space in the following way

$$\mathbf{x} \longmapsto \phi(\mathbf{x}) = \kappa(\mathbf{x}, \cdot) = \exp\left(\frac{\| \mathbf{x} - \cdot \|^2}{2\sigma^2} \right) \tag{3.47}$$

with the inner product between function given by

$$\left\langle \sum_{i=1}^{l} \alpha_i \kappa(\mathbf{x}_i, \cdot), \sum_{j=1}^{l} \beta_j \kappa(\mathbf{x}_j, \cdot) \right\rangle = \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \beta_j \kappa(\mathbf{x}_i, \mathbf{x}_j) \tag{3.48}$$

Each point can be pictured as representing a new potentially orthogonal direction, but with a higher overlap with the other directions represented by close points in the input space.

3.2.5 Kernels for structured data

As already mentioned in the previous section, kernels allow the computation of a similarity measure for arbitrary complex structures. This enables the generalization of learning tasks to these types of inputs. Here, I deal with two particular types of structured data that are particularly relevant to the scope of this thesis, i.e. strings and graphs. In computational biology it is pretty common to deal with strings that represent the sequence of molecules (e.g. RNAs and proteins). In addition, in some cases the secondary structure of RNA molecules is taken into account, and this type of information is easy to encode in graph form. This requires fast and accurate graph kernels.

In my research work I broadly employed string and graph kernels to represent RNA molecules. In Chapter 5 the RNAs are represented using a string kernel, while in Chapter 4 and 6 the RNAs are represented using a graph kernel computed over their predicted secondary structure.

3.2.5.1 Kernels on strings

The purpose of kernels on strings is to embed two sequences in a high-dimensional space where their similarity is reflected by the relative distance between the high-dimensional representations.

First, I introduce the concepts of string, substring and subsequence of symbols. The term substring refers to a string occurring contiguously within a string, while a subsequence allows the possibility that gaps separate the characters resulting in a non-contiguous instance within the string.

Definition 3.2.5 (String)**.** An alphabet is a finite set Σ of $|\Sigma|$ symbols. A string $s = s_1 \dots s_{|s|}$ is a finite sequence of symbols from Σ, including the empty sequences that is denoted by ε and it is the only string of length 0. Σ^n denotes the set of all finite strings of length n, and Σ^* stands for the set of all the strings defined on the alphabet Σ. Let s and t be strings, $|s|$ denotes the length of the string s and st is the the string obtained by concatenating s and t (of length $|st| = |s| + |t|$).

Definition 3.2.6 (Substring)**.** A string t is a substring of s if there exist two strings u and v (possibly empty) such that $s = utv$. If $u = \varepsilon$ then t is a prefix of s, while if $v = \varepsilon$ t is called suffix. For $1 \leq i \leq j \leq |s|$, the string $s(i : j)$ is the substring $s_i \dots s_j$ of s. The substrings of length k are also called k-grams or k-mers.

Definition 3.2.7 (Subsequence)**.** A string u is a subsequence of a string s, if there exist indices $\mathbf{i} = (i_1, \ldots, i_{|u|})$, with $1 \leq i_1 < \cdots < i_{|u|} \leq |s|$, such that $u_j = s_{i_j}$, for $j = 1, \ldots, |u|$, or in short $u = s(\mathbf{i})$. $|i| = |u|$ represents the number of indices in the subsequence, while the length $l(\mathbf{i})$ of the subsequence is $i_{|u|} - i_1 + 1$, that is, the number of characters of s covered by the subsequence. Conventionally, bold indices range over strictly ordered tuples of indices, belonging to the sets

$$I_k = \{(i_1, \ldots, i_k) : 1 \leq i_1 < \cdots < i_k\} \subset \mathbb{N}^k, k = 0, 1, 2, \ldots$$

All the kernels on strings presented here are explicit embedding maps from the space of finite strings defined on an alphabet Σ to a vector space F. The coordinates of F are indexed by a subset I of strings over Σ, that is a subset of the input space. Depending on the case, I can be the set Σ^p of strings of length p giving a vector space of dimension $|\Sigma|^p$, or it can be the infinite-dimensional space indexed by Σ^*. As usual, ϕ represents the feature mapping

$$\phi : s \longmapsto (\phi_u(s))_{u \in I} \in F \tag{3.49}$$

Fixed the embedding space F, there are many different maps ϕ to choose from, that will produce different feature encodings of the same strings.

One intuitive way to compare two strings is to count how many contiguous substrings (of a given length) they have in common. This simple way of comparing strings has found successful application in bioinformatics (Leslie *et al.*, 2002, 2004). The spectrum of order p (or p-spectrum) of a sequence s is the histogram of the frequencies of all its contiguous substrings of length p. Given the p-spectra of two strings, a kernel can be defined as the inner product of their p-spectra.

Definition 3.2.8 (p-spectrum kernel)**.** The feature space F associated with the p-spectrum kernel is indexed by $I = \Sigma^p$ with the embeddings given by

$$\phi_u^p(s) = |\{(v_1, v_2) : s = v_1 u v_2\}|, u \in \Sigma^p$$

and the associated kernel is defined as

$$\kappa_p(s, t) = \langle \phi^p(s), \phi^p(t) \rangle = \sum_{u \in \Sigma^p} \phi_u^p(s) \phi_u^p(t)$$

The p-spectrum kernel can be recursively computed by exploiting an auxiliary kernel, called k-suffix, and defined by

$$\kappa_k^S(s,t) = \begin{cases} 1 & \text{if } s = s_1 u,\ t = t_1 u, \text{ for } u \in \Sigma^k \\ 0 & \text{otherwise} \end{cases} \tag{3.50}$$

Then the p-spectrum kernel can be computed by

$$\kappa_p(s,t) = \sum_{i=1}^{|s|-p+1} \sum_{j=1}^{|t|-p+1} \kappa_k^S(s(i:i+p), t(j:j+p)) \tag{3.51}$$

An extension of the p-spectrum kernel considers all contiguous and non-contiguous subsequences of a string and it is named all-subsequences kernel.

Definition 3.2.9 (All-subsequences kernel)**.** The feature space associated with the embedding of all-subsequences kernel is indexed by $I = \Sigma^*$, with the embedding given by

$$\phi_u(s) = |\{\mathbf{i} : u = s(\mathbf{i})\}|, u \in I$$

that represents the number of times a subsequence u occurs in the string s. The associated kernel is then defined as

$$\kappa(s,t) = \langle \phi(s), \phi(t) \rangle = \sum_{u \in \Sigma^*} \phi_u(s)\phi_u(t)$$

The explicit computation of the all-subset embeddings requires to account for $\min(\binom{|s|}{k}, |\Sigma|^k)$ distinct subsequences of length k, becoming infeasible for all but the smallest k values. For this reason, the direct computation of the kernel function is preferred, and it can be done by recursion as follows

$$\kappa(s,\varepsilon) = 1, \tag{3.52}$$

$$\kappa(sa,t) = \kappa(s,t) + \sum_{k:t_k=a} \kappa(s, t(1:k-1)) \tag{3.53}$$

where every string contains the empty string ε exactly once. By symmetry of kernels the same recursive relation yields for $\kappa(s,ta)$.

One adaptation of the all-subsequences kernel implies the consideration of only subsequences of a fixed length p.

Definition 3.2.10 (Fixed length subsequence kernel)**.** The feature space

associated with the embedding of the fixed length subsequence kernel of length p is indexed by Σ^p, with the embedding given by

$$\phi_u^p(s) = |\{\mathbf{i} : u = s(\mathbf{i})\}|, u \in \Sigma^p$$

that represents the number of times a subsequence u occurs in the string s. The associated kernel is then defined as

$$\kappa_p(s,t) = \langle \phi^p(s), \phi^p(t) \rangle = \sum_{u \in \Sigma^p} \phi_u^p(s)\phi_u^p(t)$$

Similarly to the all-subset kernel, the fixed length subsequence kernel of length p can be recursively computed by

$$\kappa_0(s,t) = 1, \tag{3.54}$$

$$\kappa_p(s,\varepsilon) = 0, \text{ for } p > 0, \tag{3.55}$$

$$\kappa_p(sa,t) = \kappa_p(s,t) + \sum_{k:t_k=a} \kappa_{p-1}(s, t(1:k-1)) \tag{3.56}$$

where the recursion is now defined over the prefixes of the strings, but also over the length of the considered subsequences.

A more general kernel is the gap-weighted subsequences kernel. The key idea behind this kernel is still to compare strings according to the subsequences they contain, but instead of weighting all occurrences equally, the degree of contiguity of the subsequence in the input string determines how much it will contribute to the comparison. For example: the string `"gon"` is a subsequence of the strings `"gone"`, `"going"` and `"galleon"`, but for the gap-weighted subsequences kernel the occurrence in `"gone"` is more important since it is contiguous, while the occurrence in `"galleon"` is the weakest.

Definition 3.2.11 (Gap-weighted subsequences kernel)**.** The feature space associated with the embedding of the gap-weighted subsequences kernel of length p is indexed by Σ^p, with the embedding given by

$$\phi_u^p(s) = \sum_{\mathbf{i}:u=s(\mathbf{i})} \lambda^{l(\mathbf{i})}, u \in \Sigma^p$$

where $\lambda^{l(\mathbf{i})} \in (0,1)$ is the exponentially decaying weight parameter that

accounts for the gaps in the occurrence of u. The associated kernel is then defined as

$$\kappa_p(s,t) = \langle \phi^p(s), \phi^p(t) \rangle = \sum_{u \in \Sigma^p} \phi^p_u(s) \phi^p_u(t)$$

With $\lambda = 1$ the gap-weighted subsequences kernel is equivalent to the fixed length subsequences kernel. On the other hand with $\lambda \to 0$ it turns into an approximation of the p-spectrum kernel since the relative weighting of strings longer than p tends to zero. For these reasons, the gap-weighted subsequences kernel can be interpreted as a hybrid version of the other two kernels.

3.2.5.2 Kernels on graphs

Graphs are more complex structures with respect to strings. Here, I start by giving the basic definition of graphs and the associated concepts.

Definition 3.2.12 (Graph)**.** A graph $G = (V, E)$ consists of two sets V and $E \subset V \times V$. The notation $V(G)$ and $E(G)$ is used when G is not clear from the context. The elements of V are called vertices and the elements of E are called edges.

Definition 3.2.13 (Neighborhood subgraph)**.** The distance between two vertices u and v, is the length of the shortest path between them and it is denoted by $\mathcal{D}(u, v)$. The neighborhood of radius r of a vertex v is the set of vertices at a distance less than or equal to r from v and is denoted by $N_r(v)$. Given a graph G, the induced-subgraph on a set of vertices $W = \{w_1, \ldots, w_k\}$ is a graph that has W as its vertices and it contains every edge of G whose endpoints are in W. The neighborhood subgraph of radius r of vertex v is the subgraph induced by the neighborhood of radius r of v and is denoted by N^v_r.

Definition 3.2.14 (Labeled graph)**.** A labeled graph is a graph whose vertices and/or edges are labeled, possibly with repetitions, using symbols from a finite alphabet. The function that maps the vertex/edge to the label symbol is denoted by $\mathcal{L}$.

Definition 3.2.15 (Graph isomorphism)**.** Two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic ($G_1 \simeq G_2$) if there is a bijection $\phi : V_1 \to V_2$ such that for any two vertices $u, v \in V_1$, there is an edge uv if and only if there

is an edge $\phi(u)\phi(v)$ in G_2. Moreover, two labeled graphs are isomorphic if there is an isomorphism that preserves also the label information, i.e. $L(\phi(v)) = L(v)$

Since the introduction of convolution kernels in Haussler (1999), the approach based on decomposition has been the guiding principle in kernel design for structured objects such as graphs. According to the decomposition approach, a similarity function between graphs can be obtained by decomposing each graph into subgraphs and by constructing a valid local kernel between the subgraphs.

Here I present the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) (Costa and De Grave, 2010), a graph decomposition kernel that I broadly employed in my research work. NSPDK considers the decomposition of a graph into all pairs of neighborhood subgraphs of small radius at increasing distances.

Let $R_{r,d}$ be the relation that selects all pairs of neighborhood graphs of radius r whose roots are at distance d in a given graph G. More formally, $R_{r,d}(A_v, B_u, G)$ is the relation between two rooted graphs A_v, B_u and a graph G. $R_{r,d}(A_v, B_u, G)$ is true if and only if both A_v and B_u are in $\{N_v^r : v \in V(G)\}$, where A_v (B_u) is isomorphic to some N_r to verify the set inclusion, and $\mathcal{D}(u, v) = d$.

Let $\kappa_{r,d}$ be the decomposition kernel on the relation $R_{r,d}$

$$\kappa_{r,d}(G, G') = \sum_{\substack{A_v, B_u \in R^{-1}(G) \\ A'_{v'}, B'_{u'} \in R^{-1}(G')}} \delta(A_v, A'_{v'})\delta(B_u, B'_{u'}) \tag{3.57}$$

where $\delta(x, y) = 1$ if $x \simeq y$, and 0 otherwise (exact match kernel). Basically, $\kappa_{r,d}$ counts the number of identical pairs of neighboring subgraphs of radius r at distance d between two graph.

Finally the NSPDK is defines as

$$K_{r^*,d^*}(G, G') = \sum_{r=1}^{r^*} \sum_{d=1}^{d^*} \kappa_{r,d}(G, G') \tag{3.58}$$

where, for efficiency reasons, r^* and d^* are upper bounds on the radius and the distance parameter respectively.

For ensuring that relations of all orders are equally weighted regardless of

the size of the induced subgraphs the normalized version of $\kappa_{r,d}$ is considered

$$\hat{\kappa}_{r,d} = \frac{\kappa_{r,d}(G, G')}{\sqrt{\kappa_{r,d}(G, G)\kappa_{r,d}(G', G')}} \tag{3.59}$$

In Equation 3.57, NSPDK includes the exact match kernel over graphs. This is equivalent to solving the graph isomorphism problem, that is not known to be solvable in polynomial time. For this reason, NSPDK uses a fast but approximate technique to compute the exact match kernel over two finite graphs. First, a string encoding of the graphs is generated using a label function $\mathcal{L}$. Second, a unique identifier is obtained via a hashing function from strings to natural numbers. Using this approximate technique the isomorphism test between two graphs is reduced to a fast numerical identity test. On the other hand, it is not possible to ensure that there will not be cases where two non-isomorphic graphs are assigned the same identifier.

3.3 Pattern set mining

Pattern mining aims at finding useful patterns in the data. Useful patters are represented by manageable groups of patterns that together give useful insight about the data, show differences between different data sets, or can be used in classification or other common machine learning tasks. A pattern is a recurring structure that satisfies some given constraints (e.g. on the support, on the size, etc.), defined on an enumerable and discrete entities (e.g. item sets, graphs, sequences, trees, etc.). The most famous instance of pattern mining is the task of frequent item set mining, that is the problem of finding association rules between sets of items in a database of basket transactions (Agrawal *et al.*, 1993).

Nowadays, the world is witnessing a constant increase of the amount of available data. Just to give an example, in 2012 the entire world produced 1.8 Zettabytes (1.8×10^{21} bytes $= 1.8 \times 10^{9}$ GB) of data, and in 2014 the amount of produced data was 4 Zettabytes. According to Internet Live Stats, for every second of 2016 there was around 700 Instagram photos uploaded, 7 thousands tweets sent, 57 thousands Google searches, 60 thousands YouTube videos viewed, and 2.5 million emails sent. This constantly growing amount of data is reflected in both the number of available data sources, but also the size of such databases, and especially their growth rate. Therefore, the

identification of all the patterns in a database became a really impractical, if not infeasible, task. For this reason modern research focuses on finding small sets of patterns that are jointly optimal for the task at hand. This task is called pattern set mining. The main goal of this alternate formulation to the pattern mining task is to reduce the redundancy within the result set.

Several approaches for pattern set mining have been proposed. Here, I focus on description-based methods, that are unsupervised techniques that attempt at mining sets of patters that describe part of the dataset. In pattern set mining the pattern set is selected in a way that maximizes a certain optimality criterion. Description-based methods are divided in three main classes according the optimality criterion used to select the pattern sets: maximal coverage, minimum description length and maximal likelihood. Approaches based on maximal coverage define the quality of a set of patterns by how much of the data it can cover in as few patterns as possible (Geerts *et al.*, 2004; Miettinen *et al.*, 2008). Minimum description length (MDL) techniques are based on the principle that a good description should not just focus on covering/describing the data, but also take the complexity of the model (i.e. the set of patterns) into account. Famous MDL approaches are based on compression techniques (Tatti and Vreeken, 2008; Vreeken *et al.*, 2011). The last class, represented by maximal likelihood techniques, aims at finding descriptions with high likelihood. Here, patterns are specified with probabilistic models, and the methods try to find the set of patterns that maximizes the likelihood of the data (Yan *et al.*, 2005; Tatti and Heikinheimo, 2008).

In the rest of the section, I describe Boolean matrix factorization (Miettinen *et al.*, 2008) because it is employed in Chapter 6.

3.3.1 Boolean matrix factorization

In Miettinen *et al.* (2008) the authors introduce the discrete basis problem (DBP). This represents the problem of expressing a data matrix as the product of two factor matrices: one containing basis vectors that represent meaningful concepts in the data and another describing how the observed data can be expressed as combinations of the basis vectors. Classical decomposition methods usually return real-valued matrices, that are hard to interpret, especially when the original data is Boolean. For this reason, DBP formulates a matrix decomposition for Boolean data.

3.3.1.1 The discrete basis problem (DBP)

Consider an $n \times m$ binary matrix C. The rows of the matrix represent observations and the columns represent the attributes of the dataset. For instance, consider a course enrollment dataset where rows represent students and columns represent courses, and $C_{ij} = 1$ indicates that the i-th student is enrolled in the j-th course. A basis vector represents a set of correlated attributes. In the course enrollment dataset example, a basis vector corresponds to a set of courses that constitute a specialization area. The DBP formulation aims at discovering the specialization areas that are present in the dataset, and also discovering how each student in the dataset can be expressed by a combination of those specialization areas.

Let S and B be binary matrices of dimensions $n \times k$ and $k \times m$ respectively. The $n \times m$ matrix $P = S \circ B$ represents the Boolean product of S and B, i.e. the i-th row of P is the logical OR of the rows of B for which the corresponding entry in the i-th row of S is 1. In a more intuitive way, S is a usage matrix that contains information about which specialization areas appear in each observation, and B is the basis vector matrix that contains information about which courses appear in each specialization area.

The Discrete Basis Problem (DBP) is formally defined as follows.

Definition 3.3.1 (Discrete basis problem)**.** Given a binary $n \times m$ matrix C and a positive integer $k < min\{n, m\}$, find a $n \times k$ binary matrix S and a $k \times m$ binary matrix B that minimize

$$|C - S \circ B| = \sum_{i=1}^{n} \sum_{j=1}^{m} |C_{ij} - (S \circ B)_{ij}|$$

3.3.1.2 Solving DBP

DBP belongs to the class of $\mathcal{NP}$-complete problems. Therefore, the exact solution of DBP cannot be found in polynomial time with respect to the size of the input matrix C. In Miettinen *et al.* (2008) a greedy algorithm, based on Boolean matrix factorization, that approximates the solution of DBP is proposed. The basic idea is to exploit the correlations between the columns of the matrix C. First, the algorithm computes the pairwise associations between columns, and then, these associations are used as candidate basis vectors. Finally, a small set of candidate basis vectors are selected in a

Algorithm 1: DBP solver

Input: C, k, τ, w^+, w^-
Output: B, S
1 **for** $i = 1, \ldots, m$ **do**
2 $\quad \mathbf{a}_i := (\mathbf{1}(c(i \Rightarrow j, C) \geq \tau))_{j=1}^{m}$;
3 $B := [\,]$, $S := [\,]$;
4 **for** $l = 1, \ldots k$ **do**
5 $\quad (\mathbf{a}_i, \mathbf{s}) := \text{argmax}_{\mathbf{a}_i, \mathbf{s}^{n\times 1}}\, cover\left(\begin{bmatrix} B \\ \mathbf{a}_i \end{bmatrix}, [S \quad \mathbf{s}], C, w^+, w^-\right)$;
6 $\quad B := \begin{bmatrix} B \\ \mathbf{a}_i \end{bmatrix}$, $S := [S \quad \mathbf{s}]$;
7 return B and S;

greedy fashion to extract to set of k bases that represent the solution of DBP.

The DBP solver algorithm can be summarized with the pseudo-code of Algorithm 1. From now on, a row vector of a matrix M is denoted by $M_{i\cdot}$, a column vector by $M_{\cdot j}$, and a matrix entry by M_{ij}. The confidence of an association between the i-th and the j-th column is defined by

$$c(i \Rightarrow j) = \langle C_{\cdot i}, C_{\cdot j} \rangle / \langle C_{\cdot i}, C_{\cdot i} \rangle \tag{3.60}$$

where $\langle \cdot, \cdot \rangle$ is the vector inner product operation. An association between columns i and j is defined τ-strong if $c(i \Rightarrow j) \geq \tau$.

The first step is to compute an association matrix $A \in m \times m$ that contains the pool of candidate bases. An entry A_{ij} of the association matrix A is equal to one if $c(i \Rightarrow j) \geq \tau$, and 0 otherwise. Each row of A is considered as a candidate for being a basis vector, and the parameter τ controls the level of confidence required to include an attribute to the basis vector candidate (lines 1-2).

The k basis vectors to return are selected from the matrix A selecting the candidate bases that maximize the coverage of the input matrix C. Initially, B and S are empty matrices (line 3). During the iteration $1 \leq l \leq k$, the basis matrix B is updated in by setting the row $B_{l\cdot}$ to be the row $A_{i\cdot}$ in A

and the column $S_{\cdot l}$ to be a binary vector in order to maximize:

$$cover(B, S, C, w^+, w^-) = w^+|\{(i,j) : C_{ij} = 1, (S \circ B)_{ij} = 1\}| \\ - w^-|\{(i,j) : C_{ij} = 0, (S \circ B)_{ij} = 1\}| \quad (3.61)$$

where w^+ and w^- are weights that are used to reward the covering of 1's and penalize the covering of 0's, respectively (lines 4-6).

The greedy procedure presented here finds approximate solutions to DBP, but it is able to do it in polynomial (quadratic) time in the size of the input matrix C and the number of output bases. The first step of the algorithm constructs the association matrix A, and this can be done in $O(nm^2)$. Then, for selecting each of the k bases $O(nm^2)$ operations are required. Thus, solving DBP with this greedy algorithm has time complexity of $O(knm^2)$.

By definition of $\mathcal{NP}$-completeness, this polynomial greedy procedure cannot always find the exact solution of DBP. In fact, there are cases in which the algorithm is unable to find the optimal solution. One example is represented by the case in which all 1's in some basis vector occur in some other basis vectors. In these cases the algorithm is unable to find the basis vector that is contained in the other one.

Chapter 4

RNAcommender

The RNA interactome, obtained through high-throughput experimental techniques, is available for a small portion of the known human RNA binding proteins. The relevance of determining RNA-protein interactions, coupled with the still limited availability of experimental information, pave the way for *in silico* prediction of such interactions. In this chapter I introduce RNAcommender, a tool for genome-wide recommendation of RNA-protein interactions. The main purpose of RNAcommender is to suggest candidate RNA targets (transcripts) for RBPs of which the RNA binding activity has not yet been characterized or their substrates have not yet been identified. It exploits the interaction data available from high-throughput experiments performed on other proteins with similar domains. RNAcommender is a recommender system, that propagates interaction information from known RBPs to unexplored ones. It uses experimentally determined interactions and sequence information for both proteins and mRNAs, to attempt at completing the interaction map. For proteins with few known RNA targets (obtained from low-throughput assays), this consists in recommending additional interactions. For completely novel RBPs (or even presumed ones), it suggests the entire set of interactions from scratch. This *de novo* prediction task, also called cold start recommendation in recommender systems, requires to turn sequence information into appropriate features that measure the similarity between proteins and between mRNAs in terms of their binding capabilities. RNAcommender outputs a ranking of candidate RNA targets for each RBP of interest.

4.1 Related work

In Pancaldi and Bähler (2011) they employ SVMs and random forests to predict RNA-protein interactions. They represent proteins and RNAs with hundreds of different biological features extracted from the literature. Unfortunately, these features are not available for all proteins and transcripts, limiting the applicability of the method to a subset of RNAs and RBPs.

RPIseq (Muppirala *et al.*, 2011) and Wang *et al.* (2013) use sequence information to predict RNA-protein interactions. After computing RNA and protein features, based on their sequences, RPIseq applies a random forest classifier as well as SVM to predict the interactions. In RPI-seq, the RNAs are represented with the normalized frequency of the 4-mer on the nucleotide alphabet, while proteins are represented by the normalized frequency of their conjoint triad, i.e. 3-mer in a reduced 7-letter simplified alphabet representation of the RBP amino acid sequence. In this simplified representation the 20 amino acids are grouped in 7 categories according to the charge and polarity of their side chains. In Wang *et al.* (2013), the authors use a similar feature representation, i.e. the normalized frequency of conjoint triads for RBPs and of 4-mer for RNAs. Differently from RPIseq, they employ a naive Bayes (NB) and an extended naive Bayes (ENB) classifier to predict the RNA-protein interactions. Due to its independence assumption, the NB classifier is an effective and fast method, but the ENB yields better classification performance at the expense of the computational time. The ENB introduces the concept of dependency, by assuming more similar features to have a stronger correlation.

CatRapid (Bellucci *et al.*, 2011) uses physicochemical features (such as secondary structure, hydrogen bonding and van der Waals contributions) of molecules to build the interaction profiles. These profiles are used to estimate the propensity of RNA-protein interactions. CatRapid (but also RPIseq and the approach proposed in Wang *et al.* (2013)) is trained on RNA-protein interactions obtained from 3D complexes available in PDB (Rose *et al.*, 2015). Interactions acquired from 3D-resolved structures are clearly more accurate than interaction maps obtained with high-throughput sequencing approaches, but they are much harder to determine. Moreover, PDB complexes resolve only individual interactions between portions of proteins (usually one or two domains) and small fragments of RNA (with

median length of 21 nucleotides in eukaryotic cells). In general, all the above-mentioned tools are not suitable for the genome-wide prediction of RBP targets. The more recent CatRapid omics (Agostini *et al.*, 2013) extends the prediction of the RNA-binding propensity at a genome-wide scale. CatRapid omics, by precomputing the RNA features, allows (among other things) to query a protein against the complete transcriptome of different organisms. The limitation of CatRapid omics is that it does not allow to make genome-wide predictions for organisms different from the ones with precomputed RNA features.

4.2 Materials and methods

RNAcommender is a recommender system, that propagates interaction information from known RBPs to uncharacterized ones. It uses experimentally determined interactions and sequence information for both proteins and mRNAs and it attempts at completing the interaction map. RNAcommender outputs a ranking of candidate RNA targets for each RBP of interest. RNAcommender allows the computation, from sequence information only, of both RBP and RNA features, enabling the genome-wide prediction of RNA targets also for custom genomes.

In this section, first I present the dataset used in this work. Then, I explain how the protein and RNA explicit features are computed. Finally, I define the factorization model of RNAcommender.

4.2.1 Dataset

The AURA 2 database (August 2015) (Dassi *et al.*, 2014) contains a manually curated and comprehensive catalog of experimentally determined RBP-UTR interactions. UTRs are mRNA untranslated regions that are known to be highly involved in the post-transcriptional regulation.

From the AURA 2 database, I selected all the interactions obtained with high-throughput techniques, because they allow to validate RNAcommender predictions. The selection includes 67 distinct RBPs interacting with 72,226 UTRs for a total of 502,178 interactions. RBPs with high-troughput experimental evidence bind from 400 to 31,964 different UTRs, with a mean of 7,495 and a median value of 4,503, while the standard deviation is 7,711. The less selective RBPs interact with more than 40% of the UTRs while the

most selective ones bind less than 1% of the possible targets (the median is around 5-10%). The interaction information was encoded in an $n \times m$ matrix Y, where n and m are the number of RBPs and UTRs respectively: $Y_{ij} = 1$ if RBP i interacts with UTR j, and 0 otherwise.

4.2.2 RBP features

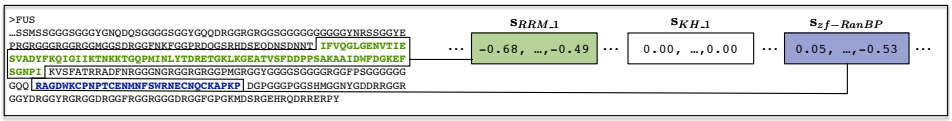
The features that represent RBPs are built using domain information included in Pfam (v. 28.0) (Finn *et al.*, 2013) because the domain information seizes affinities between protein structure, function and modularity at the same time (Lunde *et al.*, 2007).

For each RBP, its sequence is scanned against the HMM models in the Pfam-A database, selecting all the domains that matched with e-value equal to or lower than 1.0. For each protein domain found in the sequence, the Fisher score of the matching subsequence is computed. The Fisher score is obtained by computing the derivative of the subsequence log-likelihood score with respect to all the HMM model parameters (Jaakkola *et al.*, 2000). Every RBP is then represented by the concatenation of the Fisher scores of its matching subsequences with respect to their correspondent Pfam models. When multiple subsequences of an RBP are identified as the same domain, i.e. they matched the same Pfam HMM model, their Fisher scores are averaged. When a Pfam domain is not encountered in a protein a zero vector is used.

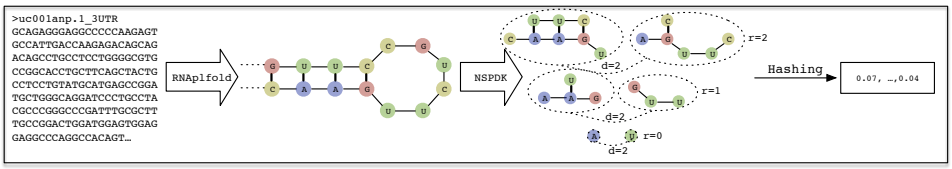
More formally, let $\mathcal{T} : \{t_1, \ldots, t_M\}$ be the set of domain types contained in Pfam (i.e. RRM_1, KH_1, ...), and $D : \{d_1, \ldots, d_N\}$ be the set of domains identified in a protein p (for example, the protein FUS incorporates an RMM_1 in position 287-365 and a zf-RanBP in position 422-453). Then, $\Theta : D \to \mathcal{T}$ is the function that maps domains of p to the domain types of Pfam. Let $D_{t_j} = \{d_i : \Theta(d_i) = t_j\}$ be the set of domains of type t_j in protein p. Let $\mathbf{s}_{d_i}$ be the Fisher score of domain d_i with respect to the HMM model of $\Theta(d_i)$. Usually, if $\Theta(d_i) \neq \Theta(d_j)$ then $\mathbf{s}_{d_i} \in \mathbb{R}^a, \mathbf{s}_{d_j} \in \mathbb{R}^b$ with $a \neq b$. The averaged Fisher score with respect to a domain type t_j is computed by:

$$\mathbf{s}_{t_j} = \begin{cases} \frac{1}{|D_{t_j}|} \sum_{d_i \in D_{t_j}} \mathbf{s}_{d_i} & \text{if } |D_{t_j}| > 0 \\ \mathbf{0} & \text{otherwise} \end{cases} \tag{4.1}$$

and the Fisher score of a protein p is the concatenation of the Fisher scores



(a) Each protein is represented by the concatenation of the Fisher scores (Jaakkola *et al.*, 2000) of its domains with respect to their correspondent Pfam models. Missing Pfam domains are represented with a zero vector.



(b) The RNA secondary structure is predicted using RNAplfold (Lorenz *et al.*, 2011), then the feature representation is computed using the NSPDK approach that extends the notion of k-mers (with gaps) from the domain of strings to the domain of graphs.

Figure 4.1: Computation of explicit features for RBPs and RNAs.

with respect to all Pfam domains: $\mathbf{s}_p = [\mathbf{s}_{t_1}, \ldots, \mathbf{s}_{t_M}]$ (Figure 4.1a).

Finally, for controlling the dimensionality of the vectors representing the RBPs, each protein is depicted in terms of its empirical kernel map, i.e. the similarity of a protein with respect to all the other RBPs. The similarity between two RBPs is estimated as the normalized dot product between their Fisher score vector representations: $sim(p,q) = \langle \mathbf{s}_p, \mathbf{s}_q \rangle / \sqrt{||\mathbf{s}_p|| \cdot ||\mathbf{s}_q||}$.

4.2.3 RNA features

The self interacting structure of an RNA sequence plays a key role in the understanding of protein binding processes. Although high-throughput protocols that allow to determine the RNA structure are now available (Sugimoto *et al.*, 2015), there is still little experimental evidence about the folding structure of full-length RNA molecules. One has still to rely on *in silico* techniques to estimate the structural behavior of such molecules from their sequence.

In Lange *et al.* (2012), different secondary structure prediction methods have been judged, concluding that local folding can be more accurate than global approaches. In order to achieve a good balance between maximizing the number of accurately predicted base pairs, and minimizing the effects of incorrect long distance predictions, they recommend a maximal span of 150 nucleotides. RNAplfold (Lorenz *et al.*, 2011) is used to estimate base

pairs probabilities, constraining interactions to spread within a maximal span, that makes suitable the scan of long RNA sequences. To the extent of considering only reliable predictions, the RNAplfold locality parameter is set to 150 nucleotides, the maximum span is reduced to 40 nucleotides and the average base pair probability cut-off is set to 0.4. Differently from sequence based approaches, here an explicit molecular graph is built, where the vertices represent by the nucleotides and the edges depict the predicted base pairs and the ribose-phosphate backbone (Figure 4.1b).

After predicting the RNA secondary structure, the Neighborhood Subgraph Pair Decomposition Kernel (NSPDK) approach, presented in Costa and De Grave (2010), is used to efficiently compute a sparse feature representation from the graph encoding. NSPDK extends the notion of counting common gapped k-mers in a string to the domain of graphs. A unique numerical identifier is given to all distinct neighborhood subgraphs using a fast hashing technique, obtaining a sparse feature encoding. Rather than considering subsequences of length k (the k-mers), NSPDK looks at neighborhood graphs of maximal radius R, that are defined as the subgraphs induced by all the vertices within a given maximal distance R from a given node. To generalize the notion of gaps, that allows components that differ in some positions to still match, NSPDK considers pairs of neighborhood graphs at a maximal distance D as a unique entity. In this way, the matching operation ignores all the vertices that are in between the two neighborhood graphs. For example, consider the case marked as r=0, d=2 in Figure 4.1b, where the 'G' intermediate node is ignored and the feature can be matched to any pair of vertices with labels 'A' and 'U' that are at a distance of 2. The full set of features is produced considering all vertices of a graph as roots and all possible combinations of radius and the distance values, up to the user defined maximal values R and D. As suggested in Heyne *et al.* (2012) both values are set to 2. The dimensionality of the feature space can be controlled adjusting the co-domain of the hashing function that turns graphs into integers. Small dimensionality values imply efficient memory footprint and subsequent processing, but also a higher risk of collision, i.e. assigning the same integer identifier to non-isomorphic subgraphs, producing a noisier encoding. In Li and König (2010) theoretical robustness guarantees have been shown when considering codes obtained from the lowest bits of each hashed value. Here, only the 10 lowest bits are considered, effectively limit-

ing the feature space dimensionality of the RNA structure encoding to 1024 (Figure 4.1b).

4.2.4 The model

The model is inspired by the matrix factorization (MF) techniques used in collaborative filtering (Koren *et al.*, 2009). Here RBPs represent the users while the items are portrayed by RNAs. MF projects both RBPs and RNAs into a latent feature space where large correlation between latent representations of an RBP and an RNA produces a recommendation. In the basic form of MF, learning aims at determining two low-rank matrices P and R such that the interaction matrix Y, can be approximated by multiplying the two low-rank matrices ($Y \approx PR^\top$). This collaborative filtering approach has proven effective to build recommender systems for movies (Koren *et al.*, 2009), but it is not applicable, as is, to this recommendation task for two main reasons. First, the unavailability of interaction information for test proteins introduces a severe cold start problem. This setting requires explicit feature representations for RNAs and, most importantly, for RBPs in order to carry out the recommendation task. Second, the number of RNAs is much higher than the one of RBPs, which makes difficult to directly project them both in a latent space of the same size.

Similarly to Ding *et al.* (2006), RNAcommender is based on trifactorization, but without orthogonality constraints. An analogous trifactorization approach has been proposed in the context of multi-relational learning (Nickel *et al.*, 2011). The key differences of the RNAcommender model are the addition of explicit feature representations, mediated by latent projection matrices, and the use of non-linear mappings.

The explicit feature representations for RBPs and RNAs are computed as described in Section 4.2.2 and 4.2.3 respectively. Then, these representations are mapped, in a non-linear fashion, into latent spaces of different sizes, where finally a third non-linear mapping associates them. The parameters of the three mappings are jointly tuned.

Formally speaking, let $F_p \in \mathbb{R}^{n \times l_p}$ and $F_r \in \mathbb{R}^{m \times l_r}$ be the matrices of the explicit feature representations of RBPs and RNAs, respectively. Let $A_p \in \mathbb{R}^{l_p \times k_p}$, $A_r \in \mathbb{R}^{l_r \times k_r}$, and $B \in \mathbb{R}^{k_p \times k_r}$ denote the three factors in the

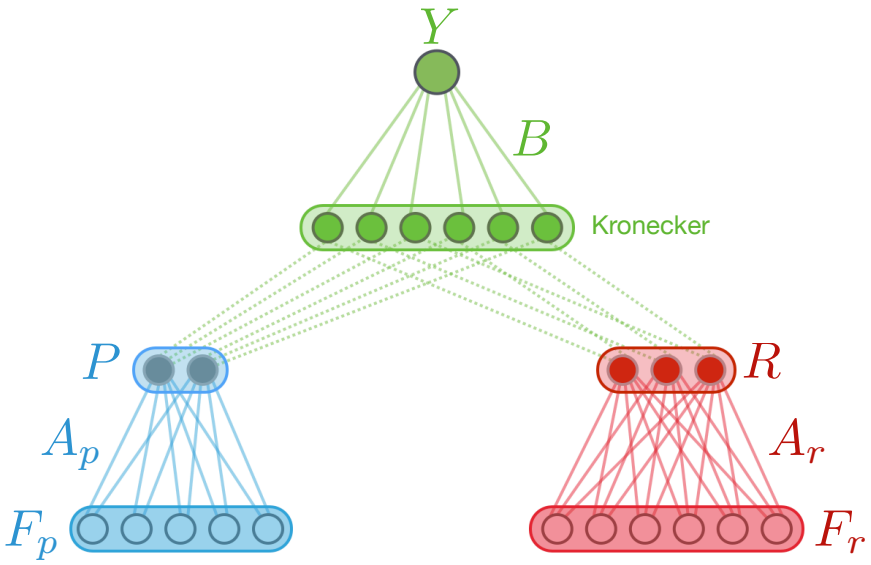


Figure 4.2: Neural network interpretation of the factorization model.

decomposition. The model is then defined by:

$$P = \sigma(F_p A_p) \in \mathbb{R}^{n \times k_p} \tag{4.2}$$

$$R = \sigma(F_r A_r) \in \mathbb{R}^{m \times k_r} \tag{4.3}$$

$$\hat{Y} = \sigma(PBR^\top) \in \mathbb{R}^{n \times m} \tag{4.4}$$

where σ is the logistic function. Alternatively, the model can be interpreted as a feedforward neural network with a Kronecker layer (second-order units) as shown in Figure 4.2. Preliminary results suggested that the use of deeper architectures, even with pretraining of the layers, increases the complexity and the model training time, without introducing significant performance improvements. Focusing on the benefit of projecting proteins and RNAs into different latent spaces, preliminary tests associated the removal of the Kronecker layer with worse recommendation performance.

The factorization model is trained using stochastic gradient descent to optimize the regularized mean squared error:

$$\min_{A_p, A_r, B} \frac{\sum_{i=1}^{n} \sum_{j=1}^{m} (Y_{ij} - \hat{Y}_{ij})^2}{n \cdot m} + \lambda \cdot r(A_p, A_r, B) \tag{4.5}$$

where $Y \in \mathbb{R}^{n \times m}$ is the interaction matrix between n proteins and m RNAs,

and the regularization term $r(A_p, A_r, B)$ is the normalized Frobenius norm of the model weights:

$$r(A_p, A_r, B) = \frac{||A_p||_F}{l_p \cdot k_p} + \frac{||A_r||_F}{l_r \cdot k_r} + \frac{||B||_F}{k_p \cdot k_r} \tag{4.6}$$

The normalization has the role of canceling out the dependency on the sizes of the model factors.

4.3 Results and discussion

For testing RNAcommender I simulated both the scenarios of predicting RNA targets for proteins on which only low-throughput analyses were performed (target completion), and the full *de novo* recommendation for proteins with no interaction information. These scenarios were simulated by masking the information of the RBPs with high-throughput information present in the AURA 2 human dataset (see Section 4.2.1). I performed leave-one-protein-out experiments, training the model on the full interaction information of $n - 1$ RBPs, and testing on the protein that was left out. In the completion setting most of the interaction information available was hidden, while in the *de novo* one I hid all the interactions. Finally, I evaluated the consistency of the model recommended RNA targets with the hidden interactions.

The tests were performed using a machine mounting 12 Intel® Xeon® CPUs E5-2603 v3 @ 1.60GHz, and 64GB of RAM, running Linux Ubuntu 14.04 LTS. Computing the features for 67 proteins took around 30 minutes (single-threaded computation), while computing the features for the 72,226 UTR sequences required 2.5 hours in multi-thread over the 12 CPUs. Training the model necessitated between 130 and 140 seconds per training epoch in multi-threaded computation over all 12 CPUs. A training epoch is defined as a complete pass over the training dataset, that, in total, contains around 4.8 million examples. I estimated that multi-threaded computation scaled the time required for training a model in an essentially linear way.

4.3.1 Protein target completion

In this section, I analyze the protein completion scenario, where the aim is to recommend RNA targets to RBPs with little interaction information

available. This situation usually occurs when the RBP interactors have been experimentally determined only through low-throughput techniques. Here I show how RNAcommender can be used to suggest targets for proteins with few known interactions, and how the introduction of the explicit features for both RBPs and RNAs can improve the recommendations.

Considering RBPs with high-throughput experiments, I assessed the performance of RNAcommender in this setting, by masking the majority of their known interactions. For each RBP, I disguised (during training) all known interactions except for 15 RNA targets. This value was estimated considering the average number of known interactions annotated in the AURA 2 database for RBPs with low-throughput evidence only. In order to obtain more reliable results, I iterated the sampling step 5 times for each RBP, and the results report mean and standard deviation.

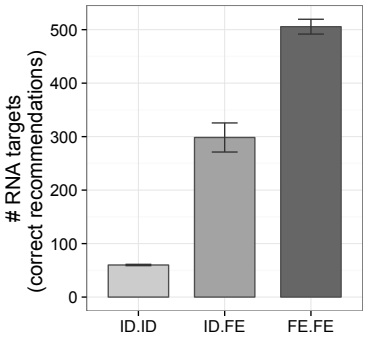
Since few known interactions of the test RBP were left in the training set, it was feasible to recommend RNA targets even without employing the explicit feature representation for RBPs and RNA targets that are presented in Section 4.2.2 and 4.2.3. Nevertheless, the results pointed out that the use of the explicit features produce better recommendations in terms of diversity and serendipity. Diversity evinces the heterogeneity of the recommended targets, measured in how many different RNA interactors are suggested when considering different RBPs, while serendipity is a measure of how surprising the successful recommendations are (Shani and Gunawardana, 2011). Here, I formalize the concept of serendipity of a recommended RNA target. For each RNA j, it is possible measure its *popularity* as the share of RBPs in the dataset binding to it: $pop_j = (\sum_{i=1}^{n} Y_{ij})/n$, where n is the number of RBPs and Y is the interaction matrix. Serendipity is inversely proportional to the concept of popularity. A common RNA, that interacts with all the proteins in the dataset, is less surprising than a target bound by only few RBPs. For this reason the *serendipity* of an RNA j is defined as $ser_j = 1 - pop_j$.

In this section, the results are reported considering three different incremental feature usage scenarios: no explicit features (ID.ID), explicit features only for the RNAs (ID.FE), and explicit features for both RBPs and RNAs (FE.FE). When explicit features were not present, the proteins and the RNAs were identified by defining $F_p = \mathbb{I}_n$ and $F_r = \mathbb{I}_m$, respectively, where $\mathbb{I}_r$ stands for the r-th dimensional identity matrix.

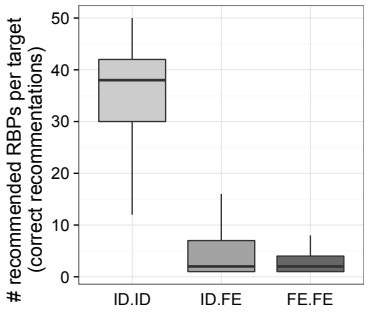
The hyperparameters of the model were optimized using a 10-fold cross-validation procedure. The obtained latent space sizes were $k_p = 5$ and $k_r = 50$. A difference in the optimal latent space sizes was expected due to the different dimensionality of the RBP and RNA sets. For the model training, the stochastic gradient descent learning rate η was set to 1.0, while the optimal value of the hyperparameter that controls the regularization of the weights, was set to different values according to the feature usage: $\lambda = 10^{-2}$ for ID.ID, and $\lambda = 10^{-4}$ for ID.FE and FE.FE. Additionally to regularization, the model also employs an early stopping approach to better deflect overfitting. The models were trained for 25 epochs for ID.ID, and 14 epochs for for ID.FE and FE.FE. By analyzing the optimal hyperparameters in the three feature usage cases, the introduction of explicit features seemed to diminish the relevance of the regularization of the model weights, and boost the convergence speed.

For each test protein, RNAcommender computes a ranking (ranging from 0 to 1) on the RNA targets. As an indicator of the overall quality the ranking, the Area Under the ROC curve (AUC ROC) for the three different feature settings was measured. Although the average AUC ROC values of the three feature usage cases were very similar: 0.76 for ID.ID and FE.FE, 0.77 for ID.FE, the diversity and serendipity of the recommended targets were rather different.

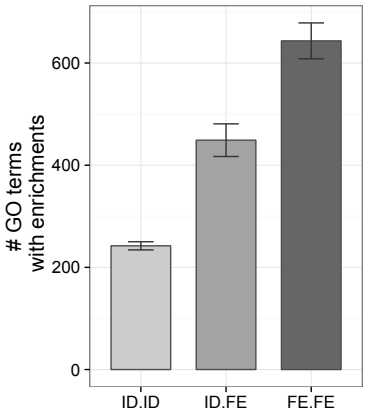
When evaluating recommender systems, it is usual to concentrate on the top recommendations because they represent the subset on which users will, most likely, focus their attention. For each test case, identified by a protein and a feature usage setting, the top 50 recommended targets were analyzed. Figure 4.3a reports the number of different correctly recommended RNA targets in the top 50 target list of at least one protein. Clearly, the introduction of explicit features increased the number of correct recommendations: from 60/68 (precision 0.88) in the ID.ID case, to 298/395 (precision 0.75) in the ID.FE case and 506/697 (precision 0.73) in the FE.FE case. Although the precision decreased, the introduction of explicit features increased the diversity and, indirectly, the serendipity because the targets for different proteins tended to be less heterogeneous. Figure 4.3b shows the box plot of the number of recommended RBPs per RNA target. Intelligibly, in the ID.ID case less differentiated recommendations were proposed: on average an RNA was recommended to 32 out of 67 proteins (with a median value of 38 RBPs).



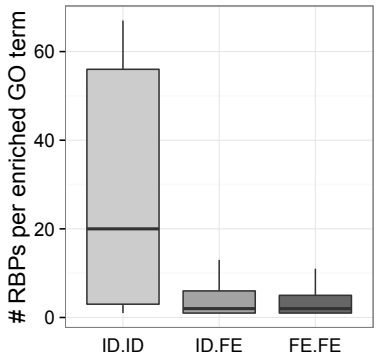
(a) Number of different targets with a correct recommendation that are included in the top 50 target list of at least one protein.



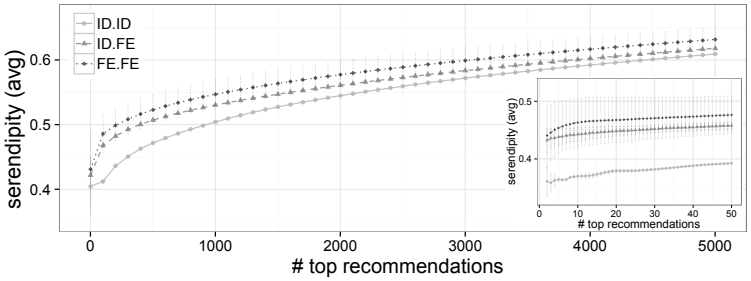
(b) Box plot of the number of recommended RBPs per RNA target.



(c) Number of different enriched GO terms associated with RNA targets that are included in the top 50 target list of at least one protein.



(d) Box plot of the number of RBPs per enriched GO term.



(e) Moving average of the serendipity of the RNA sequences along the rankings produced by the three feature settings.

Figure 4.3: Analysis of the results obtained in the low-throughput completion task.

On the other hand, the introduction of explicit features (ID.FE and FE.FE) promoted the recommendation of very diverse targets: an RNA was recommended to averagely 6 proteins in the ID.FE case and to 3 proteins in the FE.FE case (with a median of 2 RNAs in both cases). Similarly, an increased diversity and serendipity was observed when analyzing the functional enrichments of sets of predicted targets. More specific and diverse Gene Ontology enrichments were produced after introducing explicit feature representations in the model, while the ID.ID scenario promoted a homogeneous set of repeated enrichments for each analyzed RBP (Figure 4.3c and 4.3d). In order to show that the previous results were not influenced by the decision of analyzing the first 50 recommendations, in Figure 4.3e I report the cumulative moving average of the serendipity of the recommendations considering up to the first 5,000 rankings. Serendipity values were averaged over all samplings (5 per protein) of all the 67 test RBPs. Even though the serendipity (for all three cases) increased along the rankings, the introduction of explicit features (ID.FE and FE.FE) augmented the serendipity of the recommended targets, by promoting less popular RNA targets than the ID.ID case.

In summary, the results presented in this section showed how RNAcommender can be used to recommend targets to RBPs with few known interactions. Although the recommendations can be done without accounting for the explicit features, the introduction of these features for both RBPs and RNAs improved the serendipity and diversity of the recommendations.

4.3.2 De novo recommendation of protein targets

In this section, I analyze the capability of RNAcommender to suggest RNA interactors to proteins without any type of interaction information. Here I show how RNAcommender is able to successfully recommend correct RNA targets to RBPs with zero interaction information.

The experiments were performed in a leave-one-protein-out fashion, by masking all the interaction information for the hidden protein. In this setting interaction-based propagation was not possible, because no interaction information was available for the test protein. For this reason, recommendations required to be driven by feature similarity with training proteins, and therefore the recommendation task was infeasible for proteins with null similarity with all other proteins in the dataset. This reduced the number of leave-one-out experiments performed in this section from 67 to 49.

The models employed in this experimental setting were trained by using the same cross-validated hyperparameters selected for the FE.FE case in Section 4.3.1.

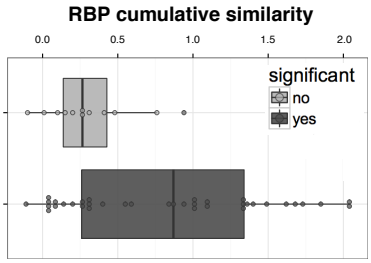
Table 4.1 reports the evaluation of the recommendations for each leave-one-protein-out experiment. Each row includes the name of the test RBP, the number of interacting RNA targets over the total of 72,226, the cumulative similarity that accounts for the similarities with the proteins in the dataset, the fraction of correct top 50 recommendations, the fraction of correct top $nTargets$ recommendations (where $nTargets$ is the number of actual targets of the test RBP) and finally the AUC ROC computed over the predicted ranking. Statistically significant enrichment in the number of correct targets in the top recommendations with respect to an equally sized random sample is represented in boldface. Statistical significance was defined by the Fisher exact test with $\alpha = 0.05$. Considering all the 49 leave-one-out experiments, the Fisher test was statistically significant in 37 cases (precision at 50). Moreover, 46 out of 49 cases were significant when considering the precision at $nTargets$. I would also like to point out that, in many cases, the p-value of the Fisher test was many orders of magnitude smaller than the significance threshold.

Similarity among RBPs and among RNAs drives the cold start recommendation. As expected, significance of the Fisher test is associated to RBPs with a high value of cumulative similarity (Figure 4.4a). Therefore, the cumulative similarity should be an aspect to take into account before attempting at recommending RNA targets for an uncharacterized RBP, because this factor seems to influence the quality of the predictions. RNAcommender learns how to weight and combine known interactions from the training proteins according to the similarities, and to use this knowledge to recommend targets of the protein of interest.

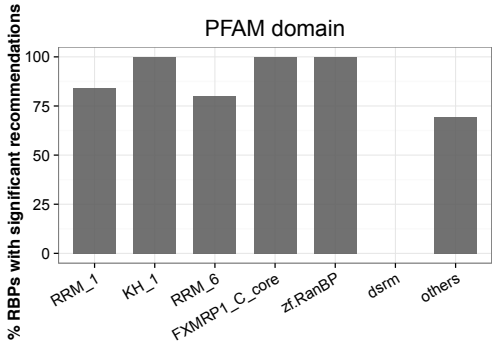
This weighted combination is supposed to be more reliable than a simpler approach as, for example, nearest neighbor. In nearest neighbor the test protein predicted targets correspond to the experimental interactions of its nearest RBP in the training set. The method showed better performance than the nearest neighbor baseline (Table 4.2). The average AUC ROC of RNAcommender was 0.75, against a value of 0.66 for the neighbor predictor. RNAcommender outperformed the nearest neighbor baseline in the majority of the comparisons, with the exception of the strongly related proteins in

Table 4.1: Evaluation of the recommendations of RNAcommender in the *de novo* setting. Test RBPs are sorted according to the precision at 50 (descending), and the number of targets (ascending). Boldface numbers indicate precisions which are significantly better than what would be obtained with an equally sized random sample according to a Fisher test ($\alpha = 0.05$).

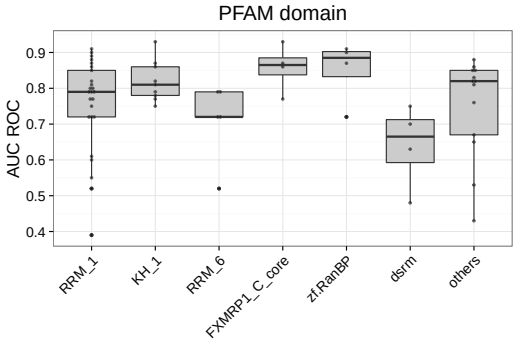
RBP	nTargets	cumSim	Pre@50	Pre@nTargets	AUCROC
TAF15	4462	1.69	**1.00**	**0.49**	0.90
FXR2	10460	1.85	**1.00**	**0.60**	0.87
LIN28B	15063	0.33	**1.00**	**0.64**	0.86
HNRNPD	15786	1.10	**1.00**	**0.41**	0.61
FMR1_iso1	16923	2.04	**1.00**	**0.66**	0.86
FMR1_iso7	18228	2.04	**1.00**	**0.58**	0.77
TIA1	19453	1.40	**1.00**	**0.73**	0.89
TIAL1	25616	1.03	**1.00**	**0.76**	0.88
AGO1	31964	0.59	**0.98**	**0.72**	0.82
EWSR1	6214	1.62	**0.96**	**0.58**	0.91
MSI1	10801	1.02	**0.96**	**0.47**	0.80
LIN28A	12821	0.33	**0.96**	**0.64**	0.88
EIF4A3	21759	0.05	**0.96**	**0.46**	0.65
RBM47	18653	−0.12	**0.92**	**0.58**	0.79
HNRNPF	4503	1.34	**0.90**	**0.30**	0.79
FUS	7577	1.74	**0.86**	**0.53**	0.87
AGO2	20761	0.40	**0.86**	**0.69**	0.85
ELAVL1	25715	1.34	**0.86**	**0.58**	0.72
DDX21	9424	0.05	**0.84**	**0.32**	0.67
ZC3H7B	12439	0.20	**0.82**	**0.51**	0.82
PCBP2	3749	0.31	**0.72**	**0.28**	0.78
FXR1	3358	1.50	**0.70**	**0.49**	0.93
YTHDF1	6648	0.26	**0.70**	**0.37**	0.81
HNRNPC	4799	0.88	**0.62**	**0.38**	0.85
RBM10	9968	0.10	**0.62**	**0.18**	0.72
HNRNPH1	4858	1.36	**0.56**	**0.23**	0.72
RBPMS	4706	0.03	**0.44**	**0.36**	0.86
IGF2BP2	9265	1.00	**0.42**	**0.40**	0.81
IGF2BP3	11429	1.15	**0.38**	**0.39**	0.75
IGF2BP1	9389	1.15	**0.30**	**0.37**	0.79
HNRNPA1	632	0.85	**0.28**	**0.18**	0.77
RBFOX2	850	0.55	**0.28**	**0.15**	0.77
HNRNPA2B1	2201	1.34	**0.28**	**0.22**	0.82
PUM2	3581	0.95	**0.18**	**0.21**	0.76
CELF1	940	0.27	**0.14**	**0.06**	0.72
QKI	1008	0.09	**0.14**	**0.12**	0.82
TARDBP	1332	0.06	**0.14**	**0.14**	0.80
STAU1	3520	0.42	0.10	**0.08**	0.48
YTHDF2	2108	0.26	0.04	**0.19**	0.85
AGO4	400	0.48	0.02	**0.04**	0.83
TARBP2	460	0.32	0.02	**0.05**	0.75
PUM1	3788	0.95	0.02	**0.12**	0.53
EIF3B	421	0.15	0.00	0.01	0.60
EIF3G	597	0.76	0.00	0.00	0.55
DGCR8	1600	0.27	0.00	**0.06**	0.63
PABPC1	2322	−0.11	0.00	**0.01**	0.39
U2AF2	2202	0.11	0.00	**0.05**	0.52
ADAR1	2210	0.02	0.00	**0.08**	0.70
RC3H1	2950	0.20	0.00	0.04	0.43



(a) Box plot of the cumulative similarity of the RBPs grouped by significance.



(b) Box plot of the number of recommended RBPs per RNA target.



(c) Box plot of the AUC ROC grouped by protein domain. The six most common domains are reported, plus "others" containing all remaining ones.

Figure 4.4: Analysis of the results obtained in the *de novo* prediction task. The six most common domains are represented separately, while the other domains are grouped together. RRM_1: RNA recognition motif 1, KH_1: KH domain, RRM_6: RNA recognition motif 6, FXMRP1_C_core: fragile X-related 1 protein core C terminal, zf.RanBP: zn-finger in Ran binding protein, dsrm: double-stranded RNA binding motif.

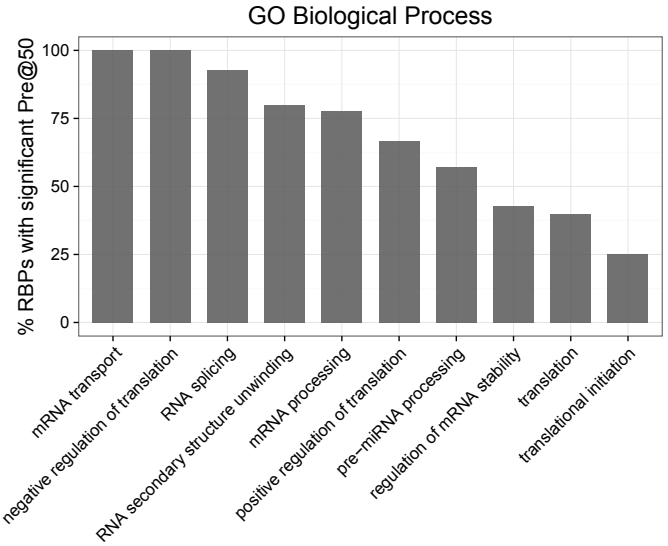
the dataset, e.g. IGFBP1, IGFBP2 and IGFBP3 or LIN28A and LIN28B.

Next, the results were analyzed in terms of domain composition of the RBPs of the dataset. Figure 4.4b reports the percentage of significant recommendations, grouping the experiments according to the test proteins domain composition, and Figure 4.4c shows the average AUC ROC values. The first six most frequent domains in the RBPs of the dataset are shown separately, while all the other domains are grouped under the category "others". As expected, the most frequent domains are RNA binding. For approximatively all the most frequent domains, the share of test proteins with significant recommendations was above 75%. The only exception is represented by proteins containing a dsrm domain (double stranded RNA binding motif): ADAR1, DGCR8, STAU1, and TARBP2. None of these proteins was significant in terms of top 50 recommendations. Even though these RBPs share a common domain type, their cumulative similarity is characterized by fairly low values (Table 4.1). The low performance might be also imputed to the quality of the training data available for these proteins. In fact, UV irradiation has a bias towards cross-linking proteins to single stranded RNAs. Thus, CLIP experiments on proteins that bind to double-stranded RNA are more likely to contain noisy information. Moreover, it is known that the interaction of the dsrm domain with RNA is unlikely to involve the recognition of specific sequences (Manche *et al.*, 1992; Polson and Bass, 1994). Still, multiple dsrm domains may be able to act in combination to recognize the secondary structure of specific RNAs (e.g. STAU1) (St Johnston *et al.*, 1992).

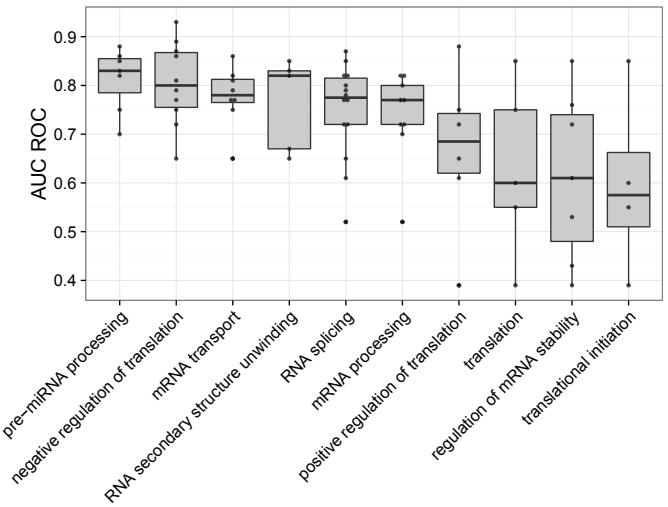
Then, I classified the RBPs with respect to their Gene Ontology annotation. Excellent performance can be observed for RBPs located in the "polysome" (CC), acting as "negative regulators of translation" or involved in "mRNA transport" (BP) and, with "mRNA binding" function (MF) (Figure 4.5, 4.6 and 4.7). Worse performance is related to translation initiation factors (EIF3B, EIF3G, PABPC1) and, again, double stranded RNA binding proteins (dsrm). Taking into account the modular behavior of molecular complexes operating in post-transcriptional gene regulation, the recommendation task should be expected to be more difficult for RBPs whose RNA interaction is mediated by other protein that belong to the complex. For example, RNAcommneder was not able to recommend any correct target in the top 50 for both EIF3B and EIF3G (see Table 4.1). They both belong

Table 4.2: Comparison with the nearest neighbor baseline. For each protein: the most similar RBP in the dataset, the AUC ROC of RNAcommender, the AUC ROC of the nearest neighbor based recommendation are shown. Nearest neighbor based recommendation policy is to suggest the targets of the most similar protein.

		AUC ROC	
RBP	**Nearest RBP**	**RNAcommender**	**Nearest RBP**
ADAR1	STAU1	**0.70**	0.51
AGO1	AGO4	**0.82**	0.51
AGO2	AGO1	**0.85**	0.77
AGO4	AGO1	**0.83**	0.75
CELF1	TIAL1	**0.72**	0.68
DDX21	EIF4A3	**0.67**	0.63
DGCR8	STAU1	**0.63**	0.50
EIF3B	TIA1	**0.60**	0.47
EIF3G	TIA1	**0.55**	0.48
EIF4A3	DDX21	**0.65**	0.57
ELAVL1	IGF2BP1	**0.72**	0.56
EWSR1	TAF15	**0.91**	0.72
FMR1_iso1	FMR1_iso7	**0.86**	0.78
FMR1_iso7	FMR1_iso1	0.77	0.77
FUS	EWSR1	**0.87**	0.73
FXR1	FXR2	**0.93**	0.91
FXR2	FMR1_iso1	**0.87**	0.86
HNRNPA1	HNRNPA2B1	**0.77**	0.62
HNRNPA2B1	HNRNPA1	**0.82**	0.54
HNRNPC	TIA1	**0.85**	0.78
HNRNPD	HNRNPA2B1	**0.61**	0.52
HNRNPF	HNRNPH1	**0.79**	0.58
HNRNPH1	HNRNPF	**0.72**	0.58
IGF2BP1	IGF2BP2	0.79	**0.90**
IGF2BP2	IGF2BP3	0.81	**0.93**
IGF2BP3	IGF2BP2	0.75	**0.86**
LIN28A	LIN28B	0.88	**0.98**
LIN28B	LIN28A	0.86	**0.93**
MSI1	FUS	**0.80**	0.60
PABPC1	HNRNPD	0.39	**0.51**
PCBP2	FXR1	**0.78**	0.54
PUM1	PUM2	0.53	0.53
PUM2	PUM1	**0.76**	0.53
QKI	PCBP2	**0.82**	0.56
RBFOX2	HNRNPC	**0.77**	0.62
RBM10	U2AF2	**0.72**	0.50
RBM47	HNRNPD	**0.79**	0.59
RBPMS	IGF2BP2	**0.86**	0.63
RC3H1	ZC3H7B	0.44	**0.49**
STAU1	TARBP2	0.49	**0.52**
TAF15	EWSR1	**0.90**	0.79
TARBP2	STAU1	**0.75**	0.68
TARDBP	HNRNPA1	**0.80**	0.52
TIA1	TIAL1	**0.89**	0.87
TIAL1	TIA1	**0.88**	0.82
U2AF2	HNRNPH1	0.52	0.52
YTHDF1	YTHDF2	**0.81**	0.63
YTHDF2	YTHDF1	0.85	**0.89**
ZC3H7B	RC3H1	**0.82**	0.50
	AVG	**0.75**	**0.66**
	STD	**0.13**	**0.15**

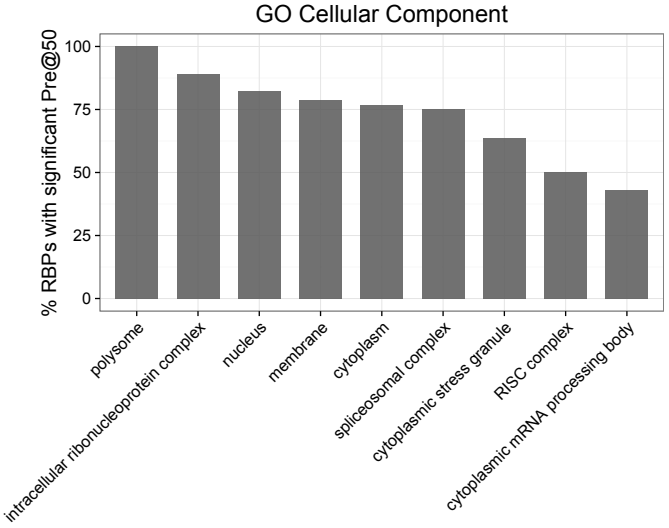


(a) For each GO category, the percentage of associated RBPs with a significant enrichment of correct predictions is displayed. GO groups are arranged from the largest to the smallest percentage.

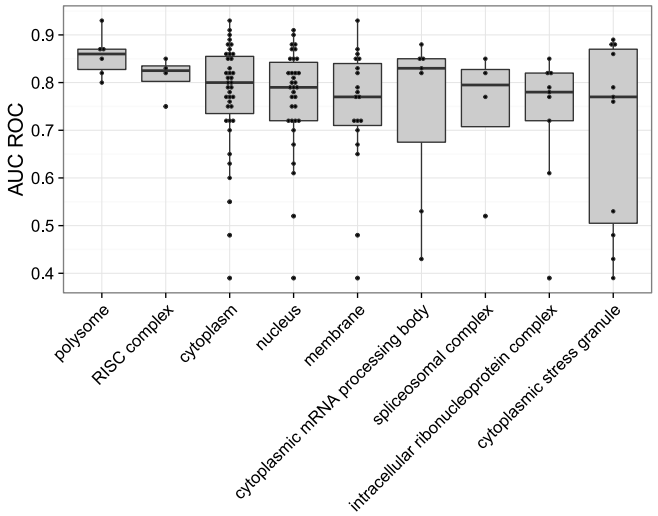


(b) For each GO category, a box plot representing the AUC ROC values of the associated RBPs is displayed. GO groups are sorted for average AUC ROC (descending).

Figure 4.5: Classification of test RBPs according to their GO annotation (Biological Process).

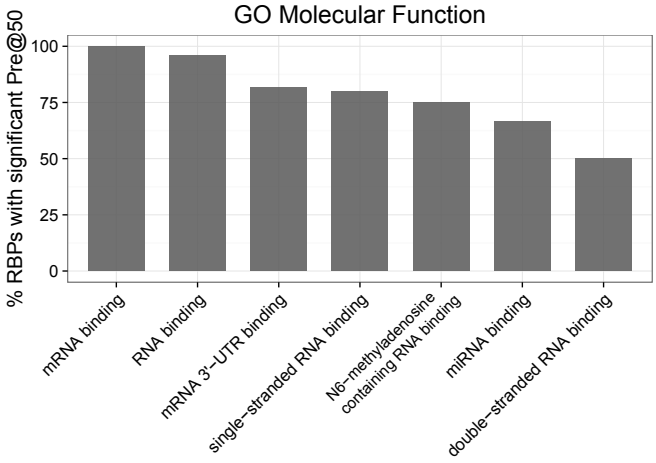


(a) For each GO category, the percentage of associated RBPs with a significant enrichment of correct predictions is displayed. GO groups are arranged from the largest to the smallest percentage.

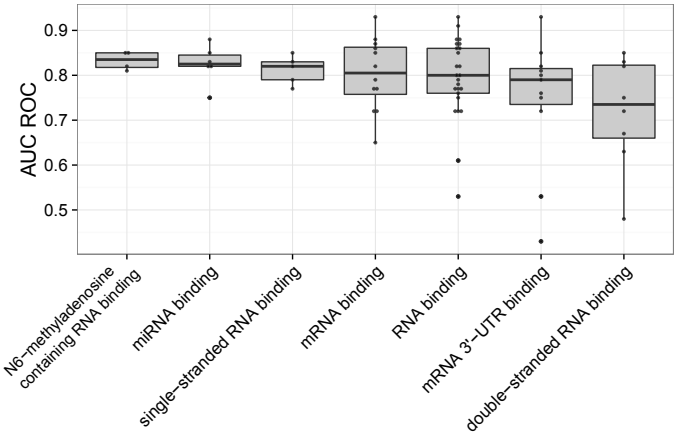


(b) For each GO category, a box plot representing the AUC ROC values of the associated RBPs is displayed. GO groups are sorted for average AUC ROC (descending).

Figure 4.6: Classification of test RBPs according to their GO annotation (Cellular Component).



(a) For each GO category, the percentage of associated RBPs with a significant enrichment of correct predictions is displayed. GO groups are arranged from the largest to the smallest percentage.



(b) For each GO category, a box plot representing the AUC ROC values of the associated RBPs is displayed. GO groups are sorted for average AUC ROC (descending).

Figure 4.7: Classification of test RBPs according to their GO annotation (Molecular Function).

to the eIF3 complex, the largest eukaryotic initiation factor, which is composed by 13 subunits (Des Georges *et al.*, 2015), and, the majority of these components are not directly interacting with the mRNA or participating in the selection of the mRNA target.

After focusing only on the top recommendations because they are reasonably the most relevant for the user of RNAcommender, I also analyzed the quality of the entire ranking produced by the tool, by measuring the AUC ROC of the recommendations. This analysis was aimed at showing that RNAcommender is able to learn an appropriate ranking of RNA targets for the test proteins. Table 4.1 also reports the value of the AUC ROC computed over the entire ranking RNA targets. High AUC ROC values were often associated to high significance of the Fisher test (e.g. TAF15, EWSR1), while AUC ROC values close to 0.5 (that corresponds to random recommendations) always corresponded to the lack of significance. However, for some test RBPs with non-significant test, the AUC ROC score was substantially better with respect to the one of a random ranking (e.g. AGO4, TARBP2). The main explanation for this result is that even though a reasonably good rank is learned, when the amount of RNA targets of the test RBP is very small it can be challenging to push them in the very top predictions. In fact, a significant fraction of correct targets, for both AGO4 and TARBP2, was found in the top $nTargets$ instead of the top 50.

Lastly, I performed a comparative analysis between the quality of the *de novo* recommendations and the FE.FE target completion task presented in Section 4.3.1. The only difference between these tasks was the number of retained interactions in the training set for the left-out protein: 15 for the protein completion case and none for the *de novo* recommendation task. This analysis was aimed at investigating whether low-throughput interaction information is actively contributing to the quality of the recommendations or not. I assessed the performance difference in terms of both AUC ROC and precision at 50. Considering that in Section 4.3.1 the 15 positive interactions were sampled 5 times for each test protein, in order to have one value per test RBP to compare with the ones reported in this section, I aggregated the performance measures by computing the median value of the 5 samples. A very small difference was registered for the mean AUC ROC value: in the case of target completion the average AUC ROC was 0.75, while in the *de novo* recommendation task it was 0.76. Also the difference in average

precision at 50 was negligible: 0.51 for the completion case, and 0.53 for the one presented in this section. The high level of correlation for both AUC ROC and precision at 50 was also confirmed by a Spearman's rank correlation of 0.98 and 0.97 respectively. These results suggest that retaining few interactions from low-throughput assays when training a model may not improve the recommendation performance. The difference in performance was not assessed for greater numbers of retained interactions in the training set, because it would be an infeasible scenario in the real world. In fact only the scenarios with no (novel proteins), few (low-throughput experiments) or all (high-throughput experiments) known interactions are meaningful in the RNA-protein interaction prediction problem.

In this section I presented an extensive analysis of the capability of RNAcommender in suggesting RNA targets to uncharacterized proteins. The results clearly indicate that, provided that the test proteins share sufficient domain similarity with other RBPs that are present in the interaction dataset, the targets of uncharacterized proteins can be predicted by the tool.

4.3.3 Recommendation for HNRNPR and SYNCRIP

Taking into account the promising results of the validation of RNAcommender performed on RBPs with high-throughput experimental evidence, the tool was used to predict the RNA interactors for RBPs lacking of such experimental evidence. As show cases I selected HNRNPR and SYNCRIP for two reasons: first, they have high similarity with other proteins with high-throughput evidence in the AURA 2 dataset; and second, the RNAcommender model trained on the high-throughput data produced recommendations with high confidence (the top 200 recommended targets received a prediction score higher than 0.99 out of 1.0 for both RBPs). The predicted rankings for the two RBPs are very similar (Spearman correlation of 0.99). This was expected considering that the two RBPs are known paralogues. The two RBPs contain almost identical RRM_1 domains, therefore RNAcommender suggested very similar targets to both RBPs.

Due to the unavailability of *in vivo* high-throughput information on these RBPs, the validation of the rankings predicted by RNAcommneder was performed using information obtained with RNAcompete (Ray *et al.*, 2009). According to the CISBP-RNA database (Catalog of Inferred Sequence Binding Preferences of RNA binding proteins) (Ray *et al.*, 2013) both

HNRNPR and SYNCRIP have high affinity with three RNA motifs. The three motifs identify all the 7-mers represented, according to IUPAC codes, by MMAAAWY, MAAAAAG and MAAAWWD. Note that MAAAAAG represents a subset of the 7-mers represented by MAAAWWD.

For each RNA target in the predicted ranking, the count of the occurrence of each possible 7-mer was estimated. In order to not bias the estimation, each count was also normalized with respect to the length of the RNA target. Then, for each 7-mer the cumulative distribution function (CDF) of the normalized counts was computed. The CDF of a 7-mer represents how the appearances of the 7-mer are distributed along the ranking. A concave CDF indicates that the 7-mer appears more frequently in the targets in the top of the ranking. Conversely, a convex CDF implies that the 7-mer appears more frequently in the bottom of the ranking.

Figure 4.8 shows the CDFs of the appearance of 7-mers in the RNA targets along the predicted rankings of HNRNPR and SYNCRIP. It is clear that the occurrences of the 7-mers with high affinity with the two RBPs (Ray *et al.*, 2013) are more frequent in the top ranked RNA targets (CDFs in red). Moreover, the comparison between the CDFs of the high affinity 7-mers (red) and the ones of all the other 7-mers (blue) shows that the high affinity 7-mers are among the most occurring 7-mers in the top RNA targets predicted by RNAcommender.

These results indicate that, without using interaction information regarding HNRNPR and SYNCRIP, RNAcommender was able to infer, from protein similarity only, the sequence affinity of the two RBPs, and to properly rank RNA targets that frequently contain such sequences.

4.4 Comparison with related work

RNAcommender proposes a new approach for predicting RNA-protein interactions. As already mentioned in Section 4.1, many *in silico* approaches have been developed to accomplish this task. In this section, I compare RNAcommender with two state of the art methods: RPIseq (Muppirala *et al.*, 2011) and CatRapid omics (Agostini *et al.*, 2013). The comparison was performed in a scaled setting that allowed the comparison with web-based services such as RPIseq and CatRapid omics.

Both RPIseq and CatRapid omics are available as web services only,

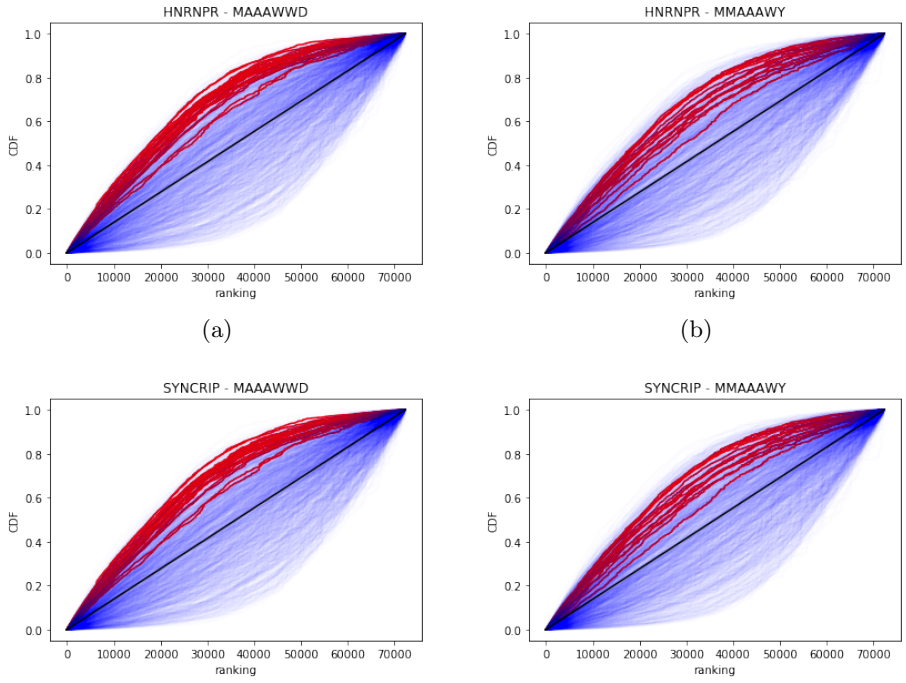


Figure 4.8: Cumulative distribution functions (CDFs) of the appearance of 7-mers in the RNA targets along the predicted rankings of HNRNPR and SYNCRIP. The CDFs of the interacting 7-mers, inferred from RNAcompete experiments (Ray *et al.*, 2013), are represented in red. The CDFs of all the other 7-mers are represented in blue. CDFs above the diagonal represent 7-mers that are more frequently present in RNA targets in the top of the rankings and less frequent in RNA targets in the bottom of the rankings.

and the imposed computational limitations denied an extensive comparative analysis with RNAcommender. RPIseq has a limit of 100 RNA sequences, while CatRapid omics has a limit of 500 sequences. However, when long RNA sequences were given as input, I estimated CatRapid's limit to be 50 RNA sequences. Depending on the length of the RNAs, CatRapid omics required from 2 to 4 hours of computation in order to generate the RNA library for 50 sequences. I also tried to generate RNA libraries for sets of 100 sequences, but in most of the cases the server timeout was reached.

Considering the mentioned limitations I was able to compare RNAcommender with both RPIseq and CatRapid testing the performance in ranking sets of 50 UTR sequences. In addition, I also compared RNAcommender

and RPIseq on sets of 100 UTR sequences. For each test RBP I sampled at random the 50 (100) UTR sequences maintaining the unbalance present in the full dataset. The reason of this choice was to reproduce the scenario in which a protein is binding a small amount of the totality of the sequences present in the dataset. In fact, on average an RBP is interacting with less than 10% of the UTR sequences in the AURA 2 dataset.

I repeated the sampling of the sequences 2 times for each test protein and the results measuring the performance in terms of AUC ROC are reported in Table 4.3 and Table 4.4. A high AUC ROC variability between the two samples can be noted for several test proteins. This effect can be imputed to the small size of the samples, meaning that the selected sequences might have strongly influenced the performance. For this reason I cannot make any statement about the protein-wise performance. However, I noted that the average performance across all the 49 test proteins was more stable, and that the average performance of RNAcommender, in the scaled settings, was very similar to the one obtained on the full dataset. Considering the average AUC ROC in this scaled setting RNAcommender seemed to outperform both RPIseq and CatRapid.

Table 4.3: Comparative analysis against RPIseq and CatRapid (50 seqeces). For RPIseq the AUC ROC is reported for both RF and SVM, for CatRapid the best AUC ROC from the predictions that considered the entire protein sequence, and the RNA-binding domains only is reported. The RNA sequences were chosen at random maintaining the same positive-negative ratio of the full dataset. Each comparison was performed on two random samples of test sequences (s1 and s2). The AUC ROC scores obtained by RNAcommneder on the full dataset are also reported (full).

	RPIseq (RF)		**RPIseq (SVM)**		**CatRapid**		**RNAcommender**		
RBP	**s1**	**s2**	**s1**	**s2**	**s1**	**s2**	**s1**	**s2**	**full**
ADAR1	0.70	0.89	0.27	0.66	0.86	0.52	0.88	0.69	0.70
AGO1	0.58	0.61	0.78	0.74	0.52	0.42	0.74	0.91	0.82
AGO2	0.77	0.57	0.83	0.66	0.46	0.48	0.88	0.81	0.85
AGO4	0.22	0.91	0.68	0.78	0.23	0.57	0.80	0.86	0.83
CELF1	0.23	0.17	0.45	0.27	0.16	0.71	0.46	0.57	0.72
DDX21	0.44	0.47	0.31	0.41	0.62	0.71	0.50	0.59	0.67
DGCR8	0.72	0.91	0.49	0.55	0.76	0.39	0.11	0.51	0.63
EIF3B	0.07	0.05	0.14	0.06	0.92	0.91	0.74	0.73	0.60
EIF3G	0.55	0.65	0.08	0.20	0.96	0.60	0.26	0.88	0.55
EIF4A3	0.58	0.44	0.47	0.64	0.58	0.58	0.64	0.71	0.65
ELAVL1	0.50	0.75	0.51	0.72	0.46	0.69	0.64	0.85	0.72
EWSR1	0.62	0.75	0.90	0.74	0.77	0.66	0.96	0.87	0.91
FMR1_iso1	0.58	0.42	0.63	0.76	0.68	0.57	0.85	0.85	0.86
FMR1_iso7	0.68	0.70	0.59	0.49	0.69	0.61	0.86	0.83	0.77
FUS	0.79	0.86	0.73	0.92	0.54	0.52	0.93	0.89	0.87
FXR1	0.55	0.58	0.92	0.81	0.73	0.60	1.00	0.95	0.93
FXR2	0.35	0.51	0.69	0.71	0.62	0.67	0.84	0.82	0.87
HNRNPA1	0.96	0.56	0.41	0.92	0.45	0.90	0.86	1.00	0.77
HNRNPA2B1	0.99	0.67	0.90	0.92	0.33	0.51	0.90	0.98	0.82
HNRNPC	0.48	0.58	0.51	0.53	0.70	0.38	0.89	0.72	0.85
HNRNPD	0.46	0.62	0.43	0.69	0.43	0.64	0.58	0.44	0.61
HNRNPF	0.79	0.53	0.94	0.38	0.24	0.69	0.81	0.72	0.79
HNRNPH1	0.80	0.48	0.42	0.37	0.60	0.68	0.61	0.80	0.72
IGF2BP1	0.74	0.56	0.82	0.41	0.34	0.64	0.91	0.66	0.79
IGF2BP2	0.71	0.69	0.68	0.71	0.65	0.59	0.85	0.90	0.81
IGF2BP3	0.60	0.55	0.55	0.68	0.65	0.61	0.65	0.74	0.75
LIN28A	0.72	0.75	0.84	0.68	0.55	0.67	0.88	0.94	0.88
LIN28B	0.69	0.82	0.57	0.71	0.77	0.54	0.83	0.74	0.86
MSI1	0.75	0.55	0.67	0.47	0.62	0.49	0.71	0.67	0.80
PABPC1	0.40	0.96	0.96	0.51	0.69	0.31	1.00	0.20	0.39
PCBP2	0.51	0.37	0.65	0.57	0.33	0.79	0.54	0.98	0.78
PUM1	0.56	0.30	0.46	0.48	0.58	0.61	0.51	0.73	0.53
PUM2	0.45	0.59	0.87	0.67	0.14	0.61	0.81	0.70	0.76
QKI	0.58	0.95	1.00	0.90	0.94	0.26	1.00	0.86	0.82
RBFOX2	0.88	0.88	0.93	0.39	0.86	0.46	0.80	0.35	0.77
RBM10	0.50	0.48	0.47	0.55	0.49	0.48	0.69	0.83	0.72
RBM47	0.71	0.66	0.88	0.68	0.44	0.40	0.86	0.87	0.79
RBPMS	0.40	0.72	0.74	0.94	0.56	0.21	1.00	0.91	0.86
RC3H1	0.18	0.81	0.25	0.42	—	—	0.02	0.46	0.43
STAU1	0.58	0.65	0.44	0.42	0.53	0.58	0.54	0.20	0.48
TAF15	0.72	0.77	0.67	0.78	0.37	0.59	0.86	0.96	0.90
TARBP2	0.73	0.80	0.86	0.82	1.00	0.62	1.00	0.82	0.75
TARDBP	0.43	0.86	0.37	0.22	0.39	1.00	0.49	0.51	0.80
TIA1	0.69	0.71	0.73	0.85	0.63	0.70	0.81	0.89	0.89
TIAL1	0.68	0.75	0.69	0.78	0.70	0.66	0.86	0.86	0.88
U2AF2	0.47	0.19	0.45	0.24	0.98	0.96	0.27	0.56	0.52
YTHDF1	0.68	0.72	0.18	0.45	0.62	0.76	0.88	0.87	0.81
YTHDF2	0.97	0.65	0.48	0.27	0.70	1.00	0.92	0.31	0.85
ZC3H7B	0.76	0.89	0.79	0.91	—	—	0.82	0.93	0.82
AVG	**0.60**	**0.64**	**0.61**	**0.60**	**0.59**	**0.61**	**0.74**	**0.74**	**0.75**

Table 4.4: Comparative analysis against RPIseq and CatRapid (100 sequences). For RPIseq the AUC ROC is reported for both RF and SVM. The RNA sequences were chosen at random maintaining the same positive-negative ratio of the full dataset. Each comparison was performed on two random samples of test sequences (s1 and s2). The AUC ROC scores obtained by RNAcommneder on the full dataset are also reported (full).

	RPIseq (RF)		**RPIseq (SVM)**		**RNAcommender**		
RBP	**s1**	**s2**	**s1**	**s2**	**s1**	**s2**	**full**
ADAR1	0.43	0.55	0.65	0.74	0.71	0.70	0.70
AGO1	0.66	0.68	0.73	0.73	0.74	0.78	0.82
AGO2	0.73	0.66	0.77	0.70	0.93	0.92	0.85
AGO4	0.53	0.72	0.23	0.82	0.83	0.99	0.83
CELF1	0.93	0.98	0.84	0.74	0.46	0.87	0.72
DDX21	0.52	0.37	0.35	0.36	0.74	0.53	0.67
DGCR8	0.66	0.54	0.49	0.91	0.50	0.85	0.63
EIF3B	0.28	0.02	0.16	0.00	0.18	0.71	0.60
EIF3G	0.27	0.78	0.68	0.53	0.37	0.61	0.55
EIF4A3	0.62	0.55	0.57	0.54	0.68	0.65	0.65
ELAVL1	0.72	0.67	0.80	0.73	0.71	0.71	0.72
EWSR1	0.81	0.63	0.87	0.69	0.88	0.94	0.91
FMR1_iso1	0.68	0.62	0.70	0.72	0.85	0.92	0.86
FMR1_iso7	0.65	0.58	0.70	0.43	0.83	0.76	0.77
FUS	0.69	0.64	0.78	0.82	0.95	0.76	0.87
FXR1	0.49	0.46	0.67	0.87	0.74	0.96	0.93
FXR2	0.56	0.73	0.76	0.70	0.92	0.88	0.87
HNRNPA1	0.89	0.99	0.99	0.86	0.80	0.97	0.77
HNRNPA2B1	0.54	0.85	0.62	0.58	0.61	0.81	0.82
HNRNPC	0.72	0.62	0.79	0.60	0.95	0.86	0.85
HNRNPD	0.63	0.46	0.62	0.50	0.67	0.46	0.61
HNRNPF	0.76	0.64	0.68	0.87	0.93	0.76	0.79
HNRNPH1	0.58	0.61	0.43	0.54	0.71	0.72	0.72
IGF2BP1	0.60	0.71	0.59	0.69	0.69	0.86	0.79
IGF2BP2	0.58	0.56	0.78	0.83	0.85	0.89	0.81
IGF2BP3	0.63	0.58	0.79	0.41	0.66	0.79	0.75
LIN28A	0.66	0.75	0.55	0.74	0.94	0.90	0.88
LIN28B	0.68	0.65	0.56	0.61	0.84	0.96	0.86
MSI1	0.86	0.66	0.75	0.69	0.90	0.81	0.80
PABPC1	0.64	0.40	0.50	0.43	0.45	0.19	0.39
PCBP2	0.59	0.67	0.60	0.59	0.73	0.53	0.78
PUM1	0.40	0.41	0.45	0.53	0.45	0.55	0.53
PUM2	0.64	0.55	0.87	0.94	0.57	0.65	0.76
QKI	0.59	0.77	0.95	0.59	0.55	0.99	0.82
RBFOX2	0.80	0.50	0.31	0.83	0.46	0.86	0.77
RBM10	0.56	0.51	0.50	0.49	0.76	0.73	0.72
RBM47	0.77	0.67	0.85	0.76	0.82	0.80	0.79
RBPMS	0.67	0.75	0.82	0.68	0.98	0.93	0.86
RC3H1	0.43	0.22	0.58	0.43	0.62	0.45	0.43
STAU1	0.38	0.36	0.20	0.49	0.52	0.63	0.48
TAF15	0.72	0.69	0.83	0.76	0.98	0.88	0.90
TARBP2	0.72	0.55	0.85	0.36	0.94	0.11	0.75
TARDBP	0.67	0.69	0.91	0.59	1.00	1.00	0.80
TIA1	0.64	0.67	0.80	0.78	0.88	0.92	0.89
TIAL1	0.73	0.62	0.80	0.80	0.94	0.87	0.88
U2AF2	0.30	0.37	0.13	0.45	0.40	0.45	0.52
YTHDF1	0.65	0.65	0.43	0.51	0.77	0.79	0.81
YTHDF2	0.65	0.53	0.39	0.28	0.65	0.90	0.85
ZC3H7B	0.70	0.67	0.82	0.77	0.89	0.78	0.82
AVG	**0.62**	**0.60**	**0.64**	**0.63**	**0.73**	**0.76**	**0.75**

Chapter 5

ProtScan

Interactions determined with high-throughput techniques are noisy and cell-line dependent. The available information is still far from being fully accurate. Due to the dependency of these techniques on expression levels and cell lines, some interactions might be missed (false negatives). Additionally, cell stress conditions, that in some cases are induced by the experimental procedures themselves, might produce some technical artifacts that are then mistakenly detected (false positives). Learning generalized models from experimentally obtained data allows to denoise the information contained in the data and to make predictions of the RBP binding preferences in conditions that are different from those used in the specific experiment. In this chapter I present a tool, named ProtScan, for precisely modeling target sites of specific RBPs using an ensemble method based on string kernels. The key idea is to cast the identification of target regions in long RNA sequences as a regression task over short moving windows, where the regressed information is the distance of the closest target site. It is well known that when trained models in an ensemble are both individually strong and collectively diverse, the consensus prediction is on average better than that of any individual trained model (Hansen and Salamon, 1990; Breiman, 2001).

5.1 Related work

Multiple sequence-motif discovery tools have been proposed to detect DNA-binding motifs of transcription factors, e.g. MEME (Bailey *et al.*, 2009) and

MatrixREDUCE (Foat *et al.*, 2006). With the increasing interest in post-transcriptional regulation these methods have been employed in or adapted to the context of RNA-protein interactions (Sanford *et al.*, 2009; Gupta *et al.*, 2013). MEMERIS (Hiller *et al.*, 2006) extends MEME including RNA accessibility information to guide the search towards single-stranded RNA regions. RNAcontext (Kazan *et al.*, 2010) considers accessibility information to define in more detail the type of unpaired regions (e.g. external regions, bulges, multiloops, hairpins and internal loops). In Kazan *et al.* (2010) a comparison between RNAcontext, MEMERIS and MatrixREDUCE showed that RNAcontext yields better performance in modeling RNAcompete data.

GraphProt (Maticzka *et al.*, 2014) employs a graph kernel, developed for RNA molecules, and SVM to learn sequence- and structure-based binding features of RNA-binding proteins from high-throughput experimental data. GraphProt improved the prediction performance in comparison to RNAcontext and MatrixREDUCE.

DeepBind (Alipanahi *et al.*, 2015) uses deep convolutional neural networks (CNNs) to model RNA-protein binding patterns from, mainly, RNAcompete data. The DeepBind approach has proven superior to several techniques on different dataset, e.g. MatrixREDUCE on RNAcompete data, and MEME on both SELEX and CHIP data.

Generally, these tools have been developed for addressing the task of the prediction of the interactivity of RBPs with RNA sequences of hundreds of nucleotides in length, and not to precisely locate the interaction sites in long RNA sequences. The only exception might be represented by GraphProt that, although being developed for the same task of the other tools, also allows to predict nucleotide-wise interaction profiles for long RNA sequences. In my opinion, all the mentioned tools are extremely important contributions to the research field, but a comprehensive tool that allows precise localization of interactions sites in entire transcriptomes was still missing. For this reason ProtScan was developed.

5.2 Materials and methods

The ProtScan pipeline is composed of several steps that can be aggregated into two main components: the first one models RNA-protein interactions and is used to predict the interaction profiles, while the second identifies

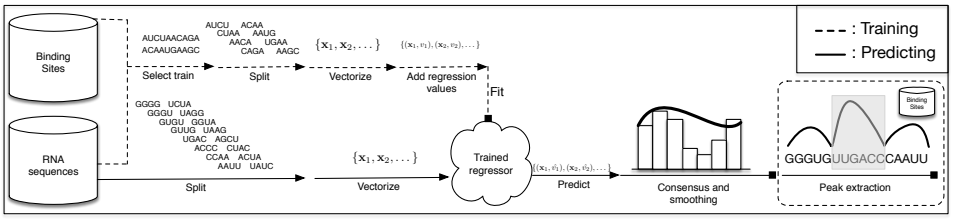


Figure 5.1: Workflow depicting the steps required for training a ProtScan model (dashed lines) and for predicting interaction profiles using a trained model (solid lines). The dashed box represents the peak extraction step.

binding sites as significant peaks in these profiles.

The first component estimates RNA-protein interaction profiles using a combination of kernelized regression with consensus voting. The kernelized regression has the task of estimating the distance of a portion of the RNA from the closest binding site, while the consensus voting is used to aggregate the predictions from the different regions. The regressor is trained using experimentally verified binding sites (dashed lines in Figure 5.1). First, an informative set of fixed-length RNA fragments was selected for the training phase. Fragments are distinguished in: positive fragments, when these are centered on a protein binding site, and negative fragments, when these are sampled at random in RNA regions that are far from binding sites. The fragments are further split into smaller overlapping windows, which are transformed into sparse vectors using a kernelized approach. Each window is annotated with its distance from the closest binding site, and a default maximal distance for the negative windows is used. Finally, a regressor is trained to predict the association between windows and their distance to the closest binding site.

In the test phase, the interaction profile for a set of arbitrarily long RNA sequences is predicted (solid lines in Figure 5.1). First, each RNA is split into small overlapping windows. The windows are then mapped to vectors, and their distance from the closest binding site is assessed using the trained regressor. All distances are then aggregated in a histogram with consensus voting. Finally, the counts are smoothed to obtain the RNA-protein interaction profile with single-nucleotide resolution.

The second component extracts the most reliable interactions from the predicted profiles. It can therefore be used to denoise a CLIP-seq experiment, removing protocol artifacts and biases. Starting from the predicted

profiles generated by the first component, ProtScan identifies all the peaks and then, using the experimental evidence as control, selects peaks above a desired significance level (dashed box in Figure 5.1).

5.2.1 Dataset

I used data obtained under the enhanced CLIP (eCLIP) (Van Nostrand *et al.*, 2016) protocol. BED narrowPeak files, containing the output of the analysis pipeline of human eCLIP experiments, were downloaded from the ENCODE project website (Sloan *et al.*, 2016) (April 2016 release). The BED narrowPeak files contain the genomic coordinates of RBP binding regions and their respective fold change values, i.e. the base 2 logarithm of the ratio between the number of aligned reads in the CLIP and the ones in the RNAseq control library. Higher fold change values are indicative of more reliable binding regions.

The full dataset includes 96 RBPs, with experiments performed in two different cell lines, i.e. K562 and HepG2 (38 RBPs on both cell lines, 40 only on K562 and 18 only on HepG2). Each experiment, identified by a protein and a cell line, was performed in two replicates. The presence of two replicates allowed to perform quality control on the data and it allows us to select only stable experiments. Binding sites are defined as regions with a fold change higher than a user-defined threshold. By setting the threshold to 2.0 and 3.0 respectively, two increasingly stringent sets of binding sites were identified. For each set, experiments where the total number of binding sites across the two replicates varies by more than 15% were discarded. A subset of 46 different RBPs passed this quality control, 8 having experiments on both cell lines, 25 only on K562 and 13 only on HepG2. When looking at the fold change threshold, 20 RBPs pass the quality control at both values, 22 only at 2.0 and 4 only at 3.0. This selection considers only eCLIP experiments that contain reasonable levels of noise, removing the RBPs for which the technique was probably unable to detect the binding sites with fair accuracy.

The BED narrowPeak files report the binding regions in genomic coordinates (hg19 assembly), but for the scope of this work the focus is on full-length gene sequences. First, genomic coordinates were converted from hg19 to hg38 assembly using the UCSC's liftOver tool (Speir *et al.*, 2016). Afterwards, genomic coordinates were converted to gene coordinates using

the human cDNA GTF file from Ensembl as a reference (release 84) (Yates *et al.*, 2016).

More formally, for each RNA sequence r the set $\mathcal{B}^r$ of coordinates b is defined, where $b = (e-s)/2$ is the center of a binding site on r that starts at coordinate s and ends at coordinate e. If an RNA sequence r has no binding sites, then $\mathcal{B}^r = \varnothing$.

5.2.2 RNA-protein interaction profiles

Here I detail the steps for the RNA-protein interaction profile estimator (Figure 5.1).

5.2.2.1 Selecting training subsequences

Training subsequences are selected in order to include information surrounding experimentally determined binding sites (positive RNA subsequences) as well as "background" information from RNA portions far away from any binding site (negative RNA subsequences). Each positive subsequence is centered on a binding site and is extended d_{max} nucleotides on both sides for a total length of $2d_{max}$. Negative subsequences have the same length but are centered on nucleotides more than d_{max} nucleotides away from the center of any binding site. Including a huge number of negatives that overwhelms the number of positives might cause improper training of the regressor. For this reason, a number of negative subsequences that is proportional to the number of positive ones (*negative_ratio* times the number of positives) are selected at random.

5.2.2.2 Splitting

Each sequence r of length l (when considering training subsequences $l = 2d_{max}$) is split in overlapping windows of size $split\_window < l$. Each window is identified by the position i of its central nucleotide on r. The amount of overlap between two consecutive windows is controlled by the parameter *split_step* with $split\_step < split\_window$ (the strict inequality ensures overlap).

5.2.2.3 Vectorizing

The splitting phase yields the instances for our regression task. A typical approach to process non-vector data (such as sequences or graphs) is to employ the kernel trick. The trick consists in using an algorithm that interacts with the input only in terms of inner product between instances. All that is needed then is a way to efficiently define an inner product between discrete sequences. A typical solution is offered by string kernels (Leslie *et al.*, 2002) that compute the fraction of common k-mers (i.e. short subsequences of length k). Here, for representational reasons, a different approach is used. An explicit feature mapping is computed from discrete sequences x to sparse vectors in very high dimensional spaces $\mathbb{R}^d$, where d is typically in the order of tens of thousands. The feature construction procedure, based on Costa and De Grave (2010), first computes $\phi_k(x) \mapsto \mathbb{R}^d$ that returns the histogram of the occurrences of each k-mer in a string x. Then, exploiting a hash function $h : \Sigma^* \mapsto \mathbb{N}$ maps k-mers (short strings in a finite alphabet Σ) to the corresponding integer codes $n \in \mathbb{N}$ in the addressable space (i.e. $n < d$). In order to take into account the contribution of k-mers of different complexities (different values of k) in a balanced way, the normalized version is considered: $\hat{\phi}_k(x) = \phi_k(x)/\sqrt{< \phi_k(x)\phi_k(x) >}$, then the vector representations for different orders k are combined in a single vector: $\phi_C(x) = \sum_{k=0}^{C} \hat{\phi}_k(x)$ and finally we output the normalized result: $\hat{\phi}_C(x) = \phi_C(x)/\sqrt{< \phi_C(x)\phi_C(x) >}$. The maximum k-mer size C, is called *complexity* of the vectorization.

5.2.2.4 Regression

In ProtScan a ridge regressor with squared loss and l_2 regularization is employed. The training of the regressor is performed using stochastic gradient descent (SGD). Let i be the center of a window of a RNA sequence r, then v_i is the corresponding regression value which is inversely proportional to the distance of i from the closest binding site on sequence r, if i is a positive window, and zero otherwise as defined by

$$v_i = \begin{cases} \max(0, 1 - \frac{\min_{b \in \mathcal{B}^r} |i-b|}{d_{max}}) & \text{if } \mathcal{B}^r \neq \emptyset \\ 0 & \text{otherwise} \end{cases} \tag{5.1}$$

In the prediction step, the distance values for RNA windows of test RNA sequences are estimated. For each test window i, the regressor predicts a

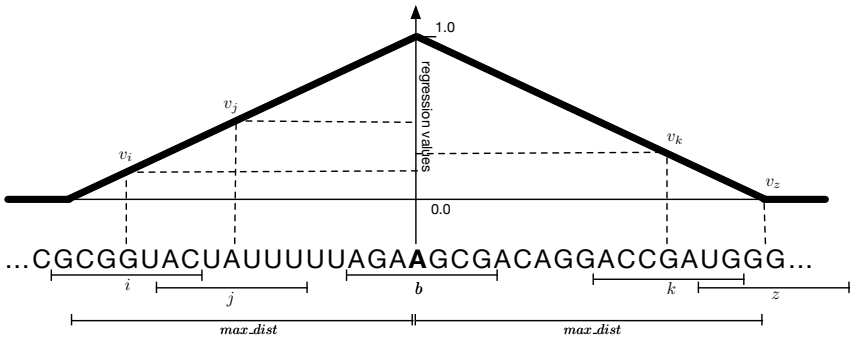


Figure 5.2: Example of the definition of regression values. The regression value v_i is lower than v_j because chunk i is farther from the target site b than chunk j. Although chunk k is positioned upstream from the binding site b, and chunks i and j are downstream, $v_i < v_k < v_j$ because the regression values do not need to account for the relative position w.r.t. the binding site but only for the absolute distance. Moreover, $v_z = 0$ because z is at d_{max} nucleotides from the center of the binding site b.

value $\hat{v}_i$. The predicted value is mapped to a distance $\hat{d}_i \in [0, d_{max}]$ inverting Equation 5.1:

$$\hat{d}_i = d_{max} * (1 - \hat{v}_i) \tag{5.2}$$

Values in $[0, 1]$ express the proximity to a binding site, where larger values indicate a closer location. Note that Equation 5.1 assigns regression values according to the absolute value of the distance from the most adjacent binding site (Figure 5.2) and that it cannot recover the relative position of the window with respect to the binding site (i.e. downstream or upstream). Encoding directionality information using, for example, negative regression values to indicate upstream locations yielded poor performance due to the discontinuity at zero. As shown below, the exact location can be recovered using a consensus voting procedure.

5.2.2.5 Consensus voting and smoothing

In test phase the predictions from all available windows are aggregated. ProtScan builds a histogram $\mathbf{h} = (h_1, \ldots, h_l)$, where l is the length of a test RNA sequence r and h_j aggregates the votes received by its j-th nucleotide. A window i is discarded if $\hat{v}_i \leq 0$ as it is predicted to be too far from a binding site to be relevant. Otherwise, every prediction contributes two votes, one upstream to position $i - \hat{d}_i$ and one downstream to $i + \hat{d}_i$ (recall

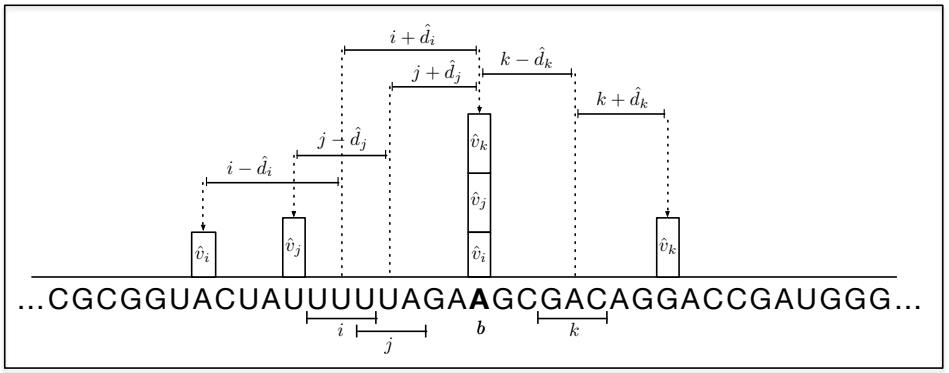


Figure 5.3: Example of the consensus voting approach. Under the assumption that the regressor is perfectly trained ($\hat{v}_c = v_c$ for all windows c), $\hat{d}_c$ represents the exact distance of a window c to the closest binding site b. From the example it is possible to notice that the votes are correctly piling up on the binding site and spreading on the other nucleotides.

that the regressor is trained over the absolute value of the distance). Votes for position j are thus computed as:

$$h_j = \sum_{i \in windows(r)} \begin{cases} \hat{v}_i & \text{if } i \pm \hat{d}_i = j \text{ and } \hat{d}_i < d_{max} \\ 0 & \text{otherwise} \end{cases} \qquad \forall j \qquad (5.3)$$

Note that in Equation 5.3 each vote is weighted according to the predicted distance, i.e. the closer the voting window the higher the weight. This is done to impose a bias whereby RNA windows that are closer to a binding site are considered more important for the protein recognition than more distant windows. Secondly, the vote is added to both $i \pm \hat{d}_i$, i.e. upstream and downstream from the window coordinate. At first glance, this seems an issue, as one of the two votes is clearly wrong. However, votes will combine in a constructive way only on the true location while they will incoherently spread out in the other direction (see Figure 5.3).

Finally, Gaussian smoothing, i.e. the convolution of histogram $\mathbf{h}$ with a Gaussian $\mathcal{N}(\mu, \sigma)$, is applied to the histogram $\mathbf{h}$ to denoise it and to produce a single-nucleotide resolution interpolated profile.

5.2.2.6 Computational efficiency analysis

Training a ProtScan model or use a model to predict the binding profile of an RBP over a set of RNAs yield a rather different complexity.

During training all the RNA subsequences, of size $2d_{max}$, centered on a protein binding site are used, plus *negative_ratio* times the number of binding sites negative RNA subsequences. Negative RNA subsequences have the same size of the positive ones, i.e. $2d_{max}$. Let n be the number of biding sites of a protein, then in total the training of a model is performend using $n * (1 + negative\_ratio)$ RNA subsequences of length $2d_{max}$. For each RNA sequence the splitting procedure generates $\frac{l - split\_window}{split\_step} + 1$ windows, therefore the number of produced windows is

$$n * (1 + negative\_ratio) * \left(\frac{2d_{max} - split\_window}{split\_step} + 1 \right) \tag{5.4}$$

These windows are then vectorized and fitted into the regressor, where the vectorization is the time consuming step.

During prediction, all the windows of all the RNAs in the test set require to be generated. Let m be the number of RNAs in the test set and l_i the length of the i-th RNA, then the number of generated windows is

$$\sum_{i=1}^{m} \left(\frac{l_i - split\_window}{split\_step} + 1 \right) \tag{5.5}$$

These windows are then vectorized and fitted into the regressor, where the vectorization is the time consuming step.

When entire genomes of complex organisms (e.g. *Homo sapiens*) are considered, usually $n * (1 + negative\_ratio) \ll m$ and $(2d_{max}) \ll l_i$. This implies that predicting the binding affinities for an entire genome might require up to 100 times the computation time required to to train a model on the same genome (this estimation has been done with the default hyperparameters of ProtScan showed in Table 5.1). The efficiency of the training procedure allows to easily train customized ProtScan models on common multi-core machines. Using these models to test relatively small sets of RNA sequences can also easily be achieved with limited computational resources. While for the prediction of entire genomes it is advised to exploit the high level of parallelization of ProtScan that allows to scale the computation, in

an almost linear fashion, over numerous CPUs.

5.2.2.7 Hyperparameter optimization

ProtScan exhibits a relatively large set of hyperparameters, that, jointly, guide the overall behavior of the model. The splitting hyperparameters, *split_window* and *split_step*, control the size and number of splits generated by the sliding window over an RNA molecule. The regression step necessitates to: assign regression values, that are dependent on the maximum distance allowed for a window to be considered close to a binding site (d_{max}); vectorize RNA windows, using the string kernel guided by the *complexity* hyperparameter that controls the maximum size of the considered k-mers; and fit the SGD regressor itself (7 more hyperparameters). Finally, the smoothing step is guided by 2 hyperparameters: the mean μ and the standard deviation σ of the Gaussian signal. An additional hyperparameter (*negative_ratio*), used only in the training step, defines the amount of negative subsequences to consider when training the regressor.

ProtScan hyperparameters are optimized using a two-fold cross validation random search approach (Bergstra and Bengio, 2012). Running the hyperparameter optimization over 34 models for 11 different RBPs, I noted that several optimal hyperparameter values where stable for a wide range of RBPs. These stable parameters have been incorporated as default parameters and allow to train ProtScan, skipping the computationally expensive hyperparameters optimization phase while maintaining high predictive performance. The full list of the set default hyperparameters is shown in Table 5.1.

5.2.3 Peak extraction

Predicted interaction profiles consist of single-nucleotide resolution signals indicative of the RNA-protein coupling. However, the localization of significant peaks in these profiles is a non-trivial task, akin the process of peak calling in CLIP-seq data analysis. Therefore, ProtScan includes an approach to find significant peaks and thus sites likely bound by the RBP from the predicted interaction profiles.

All the peaks in the predicted profiles are extracted using a variant of the mean shift algorithm (Comaniciu and Meer, 2002). Mean shift scans

Table 5.1: ProtScan default hyperparameters.

Context size	d_{max}	54
Preprocessing	*split_window*	70
	split_step	3
	negative_ratio	4.3
Vectorizer	*complexity*	3
SGD regression	*loss*	squared
	penalty	l2
	alpha	0.0001
	l1_ratio	0.5
	n_iter	5
	*eta*0	0.01
	power_t	0.25
Smoothing	μ	148
	σ	48

a sequence with a fixed-length sliding window and records the maximum value found in each window. It then iteratively repeats the procedure over the sequence of maxima found until no further change occurs. An analogous procedure is used to localize all the minima. After identifying all the local maxima and minima in the profile, a candidate predicted binding site is defined as a block $b = (s, e)$ with coordinates $(s, e) : s < e$. If both s and e are minima, a block contains no other minimum and at least one maximum.

In order to select the subset of significant binding sites among the extracted peaks, they are compared with a background distribution fit on negative data. First, a cumulative Gaussian distribution for the maximum is fit over the height of the blocks coming from transcripts without experimental evidence of binding (negative examples). Second, each candidate block is accepted as significant if it stays in the top θ^{th} percentile of the distribution, with θ specified by the user. The procedure is cross validated two-fold to avoid overfitting.

5.3 Results and discussion

In this section I analyze the potentiality of ProtScan to model and predict RNA-protein interaction profiles at a transcriptome-wide scale.

5.3.1 Transcriptome-wide target site modeling

ProtScan can be used to model RNA-protein interactions and to predict interaction profiles at a transcriptome-wide scale. Here I present the results that show how ProtScan is able to predict binding sites regions and that the ProtScan models effectively model RBP biding preferences.

As examples I considered the two vastly studied RBPs HNRNPA1 and FMR1. These RBPs are of broad interest because of their involvement in different cell diseased states (Richter *et al.*, 2015; Geuens *et al.*, 2016). Note that these RBPs act in different cellular compartments, i.e. the nucleus for HNRNPA1 and the cytoplasm for FMR1. Nuclear RBPs, especially splice factors such as HNRNPA1, interact with pre-(m)RNA that is composed of introns and exons, while cytoplasmic RBPs such as FMR1 interact with mature RNA molecules, from which the intronic sequences have been removed during splicing. Dealing with mature RNAs and ignoring intronic sequences shortens the computation time required for predicting the binding profiles of an order of magnitude.

HNRNPA1 is part of a family of ubiquitously expressed heterogeneous nuclear ribonucleoproteins (hnRNPs). These RBPs are known to associate with pre-(m)RNAs in the nucleus and influence their processing, as well as other aspects of RNA metabolism and transport. HNRNPA1 is one of the most abundant core proteins of hnRNP complexes and plays a key role in the regulation of alternative splicing. Mutations in the HNRNPA1 gene have been observed in individuals with amyotrophic lateral sclerosis (ALS) (Geuens *et al.*, 2016). Here the eCLIP experiment on K562 cells (replicate 1) was considered. The binding sites were selected using a fold change threshold of 2.0, resulting in 4,964 interaction sites. The interaction profiles for the entire set of human genes was predicted using a two-fold cross-prediction procedure analogous to the one employed for peak extraction (using in turn one subset for training and the other for prediction), obtaining an overall AUC ROC of 0.85.

Next, the significant peaks from the predicted interaction profiles were extracted using the method proposed in Section 5.2.3. The target regions identified by ProtScan were visualized by running a motif finder procedure on the 5,000 peaks with the lowest p-value. An *in vitro* study by Burd and Dreyfuss (1994) identified the motif UAGGG(A|U) as a consensus high affinity HNRNPA1 binding site. This consensus sequence is well represented in

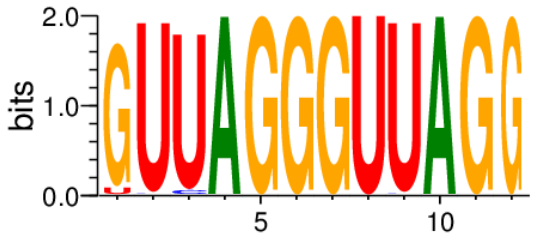
the HNRNPA1 motif displayed in Figure 5.4a. The 12-mer GUUAGGGUUAGG occurred 63 times (exact match) in the analyzed subsequences.

Differently from HNRNPA1, FMR1 is known to associate with polysomes, and an expansion of the CGG repeat in the 5' UTR of the FMR1 gene is known to cause the fragile X syndrome (FXS) (Richter *et al.*, 2015). Here the eCLIP experiment on K562 cells (replicate 1) was considered. The binding sites were selected using a fold change threshold of 2.0, resulting in 26,732 interaction sites. The fact that FMR1 is usually located at polysomes in the cytoplasm allowed to consider only mature RNAs, i.e. RNAs without intronic sequences. In humans, alternative splicing enables the production of more than one transcript from each gene. In order to not consider every splice variant of each gene, the most prominent transcript was selected through a series of hierarchical filtering steps: first the transcript support level (TSL) that identifies well supported transcripts was considered[1], then the APPRIS annotation (Rodriguez *et al.*, 2015) that annotates principal splicing isoforms, followed by the GENCODE basic annotation that identifies the representative transcripts of a gene, and finally the transcript length (preferring longer transcripts). If the procedure ended up producing two or more transcripts (which are on par on all parameters), the most prominent transcript was selected at random among them. The selection of the most prominent transcript for each gene allowed to significantly reduce the size of the dataset and, therefore, to speed up the prediction of the interaction profiles for this RBP. Cross-predicted interaction profiles achieved an AUC ROC of 0.79.

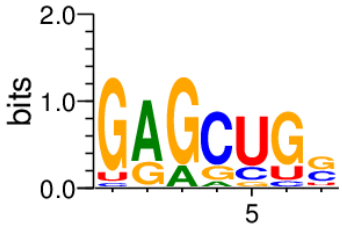
As with HNRNPA1, the FMR1 target regions obtained from the analysis of the 5,000 peaks at lowest p-value were visualized. A PAR-CLIP study of FMR1 target sites (Ascano *et al.*, 2012) identified two distinct motifs for this RBP: ACUG and UGGA. These motifs are in substantial agreement with those extracted from the ProtScan profiles. The 7-mer GAGCUGG (Figure 5.4b) occurred 445 times (exact match) in the considered subsequences, while the 6-mers matching the following regular expression (C|G)(C|U)(G|U)G(G|A)(A|G) (Figure 5.4c) were found 4233 times.

The sufficiently high AUC ROC scores indicate that ProtScan can be

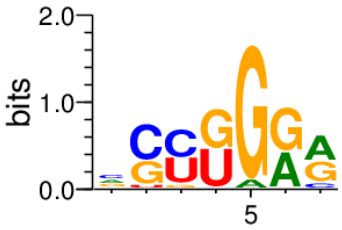
[1]The Transcript Support Level (TSL) is a method to highlight the well-supported and poorly-supported transcript models for users. The method relies on the primary data that can support full-length transcript structure: mRNA and EST alignments supplied by UCSC and Ensembl.



(a) Motif for HNRNPA1.



(b) First motif for FMR1.



(c) Second motif for FMR1.

Figure 5.4: Motifs for HNRNPA1 and FMR1.

used to reliably model the interaction profiles on a transcriptome-wide scale. The agreement between the resulting motifs and those identified in *ad hoc* studies (Burd and Dreyfuss, 1994; Ascano *et al.*, 2012) further supports the quality of the predicted interaction profiles.

In summary, these results showed the capability of ProtScan of operating at a genome-wide scale. The tool was able to accurately predict RNA binding sites for two RBPs of broad interest that are localized in different cell compartments, i.e. nucleus and cytoplasm. Moreover, ProtScan was able to model the sequence preference of both RBPs.

5.4 Comparison with related work

I compared ProtScan with GraphProt (Maticzka *et al.*, 2014) and DeepBind (Alipanahi *et al.*, 2015). Although both approaches have proven superior to several state-of-the-art methods on different types of interaction data (e.g. CLIP, RNAcompete, SELEX, CHIP), a comparison between the two, performed on interactions obtained *in vivo*, was still lacking. For this reason I compared ProtScan with these two approaches.

Both GraphProt and ProtScan are based on less complex models than the deep CNNs used in DeepBind. Therefore training their models requires

significantly less time in comparison to training deep CNNs employed in DeepBind. In fact, training a DeepBind model, from RNAcompete data, necessitates powerful hardware such as a GPU cluster. While GraphProt and ProtScan models can be trained on human high-throughput data in few hours on a common multi-core machine, DeepBind delivers the pretrained models together with the software. On the other hand, trained DeepBind models are fast in producing predictions because they only require the forward pass through the traned neural network. This leads to a 10 to 20 times faster testing procedure compared to GraphProt and Protscan, that need to compute the features for the test RNA sequences.

Due to the inability of training custom DeepBind models, the three methods were compared using the RBPs that are present in our dataset (Section 5.2.1) and that have a pretrained DeepBind model. This lowered the number of RBPs usable for the comparison to 11. For each protein, multiple tests were performed considering different cell lines, fold change values (to define binding sites from experimental evidence), and technical replicates, for a total of 34 comparisons. The performance of the three approaches was analyzed on $\sim$ 1% of human genome ($\sim$ 600 protein coding and non-coding genes). The test genes were selected at random, keeping the same ratio between bound and unbound genes that is present in the full dataset.

For each RNA molecule in the test set, a vector of interaction scores was computed, one score per nucleotide, representing the strength of the predicted interaction with the test RBP. The interaction profiles of ProtScan were computed as explained in Section 5.2.2.

Although GraphProt is mainly built to discriminate between interacting and non-interacting RNA stretches, it also allows to predict an affinity profile for an entire RNA molecule returning one score per nucleotide. The score of each nucleotide is equal to the margin of the SVM classifier obtained with the feature representation of the nucleotide. Usually, Graphprot represents an RNA sequence with the features generated considering all the gapped k-mers (or pairs of neighborhood subgraphs when the secondary structure of the molecule is taken into account). Instead, the feature representation of a nucleotide is obtained by considering only the features generated by the k-mers (or pairs of neighborhood subgraphs) that include the nucleotide.

Differently from GraphProt, DeepBind does not allow to predict inter-

Table 5.2: Performance comparison among GraphProt (Maticzka *et al.*, 2014), DeepBind (Alipanahi *et al.*, 2015) and ProtScan considering 11 RBPs. For each test protein multiple tests are performed taking into consideration different cell lines (CL), fold changes (FC), and replicates (R), for a total of 34 comparisons. For each comparison, the best score is highlighted in boldface.

				AUC ROC		
RBP	**CL**	**FC**	**R**	**GraphProt**	**DeepBind**	**ProtScan**
FMR1	K562	2.0	1	0.83	0.63	**0.88**
			2	0.80	0.62	**0.84**
GTF2F1	HepG2	2.0	1	0.71	0.56	**0.80**
			2	0.78	0.58	**0.86**
		3.0	1	0.72	0.56	**0.79**
			2	0.80	0.58	**0.86**
HNRNPA1	HepG2	2.0	1	0.72	0.76	**0.81**
			2	0.72	0.75	**0.82**
	K562	2.0	1	0.72	0.77	**0.80**
			2	0.72	0.74	**0.83**
		3.0	1	0.71	**0.77**	**0.77**
			2	0.72	0.74	**0.80**
HNRNPC	HepG2	2.0	1	0.68	**0.86**	**0.86**
			2	0.71	0.77	**0.87**
HNRNPK	K562	3.0	1	0.81	0.86	**0.89**
			2	0.79	0.85	**0.90**
IGF2BP2	K562	2.0	1	0.75	0.29	**0.80**
			2	0.76	0.31	**0.79**
IGF2BP3	HepG2	2.0	1	0.66	0.35	**0.85**
			2	0.70	0.35	**0.82**
KHDRBS1	K562	2.0	1	0.60	0.64	**0.64**
			2	0.62	0.63	**0.66**
QKI	HepG2	2.0	1	0.59	0.68	**0.74**
			2	0.54	0.56	**0.74**
		3.0	1	0.58	0.68	**0.72**
			2	0.54	0.56	**0.69**
TARDBP	K562	2.0	1	0.71	0.84	**0.88**
			2	0.72	0.86	**0.88**
U2AF2	HepG2	2.0	1	0.59	0.68	**0.76**
			2	0.59	0.68	**0.72**
	K562	2.0	1	0.59	0.69	**0.78**
			2	0.62	0.67	**0.79**
		3.0	1	0.59	0.69	**0.76**
			2	0.62	0.67	**0.77**
			AVG	**0.69**	**0.65**	**0.80**
			STD	**0.08**	**0.15**	**0.07**

action profiles returning one score per nucleotide, but instead it returns one score that indicates the overall interaction propensity of the entire RNA stretch. For this reason, predicted interaction profiles for entire RNA molecules were produced using DeepBind predictions together with a sliding window approach similar to the one proposed in ProtScan. Given a transcript a sliding window procedure was applied to create many overlapping RNA windows. Each window was scored using DeepBind and the corresponding score was added to all the nucleotides belonging to the window. By applying this procedure, a histogram that represents the interactivity of each nucleotide in the transcript was obtained. Finally, a smoothing procedure was applied to obtain a continuous signal. The sliding window, and smoothing steps employed here were identical to the ones of ProtScan (see Section 5.2.2.2 and 5.2.2.5). Moreover, for splitting the transcripts and smoothing the prediction histograms, the same hyperparameters of ProtScan (see Table 5.1) were used. Therefore, the only difference between the affinity profiles predicted by ProtScan and the ones of DeepBind, was the utilization of the RNA windows: ProtScan used them to predict the position of the closest binding site, while DeepBind evaluated the interactivity of each RNA window.

For each test case, Table 5.2 reports the results in terms of AUC ROC. Also in Maticzka *et al.* (2014) and Alipanahi *et al.* (2015) the authors report the results in terms of AUC ROC, but they address a different task with respect to the one analyzed in this comparison. Both GraphProt and DeepBind have been tested on classifying whether RNA subsequences contain an RBP interaction site or not, while here the ability of the methods in localizing the binding sites on full RNA transcripts is tested. This is a much harder task for mainly two reasons. First, the amount of interaction scores to account for is significantly higher: from one score per RNA sequence (or subsequence) to one score per nucleotide. Second, when considering entire RNA transcripts, the fraction of interacting nucleotides is usually much inferior than the one of non-interacting ones, leaving a very small margin of error if good performance wants to be achieved. Just to give an example, in the dataset described in Section 5.2.1 the average ratio between interacting and non-interacting nucleotides is 1 to 2500.

By showing a best AUC ROC score in all the cases (except two cases in which it ties with DeepBind), ProtScan seems to outperform the com-

petitors. With an average AUC ROC of 0.8, ProtScan introduces a relative AUC ROC improvement of 35% over GraphProt, and of 43% over DeepBind. Analyzing the pairwise comparison of the methods, ProtScan yields superior performance than GraphProt in all 34 cases, and 32 out of 34 against DeepBind (2/34 are ties). Although GraphProt has a superior average AUC ROC than DeepBind, the latter shows better performance than GraphProt in 24 out of 34 cases. This is mainly due to the fact that the average AUC ROC of DeepBind is critically penalized by the scores obtained for the proteins IGF2BP2 and IGF2BP3. In Hafner *et al.* (2010) the authors hypothesized that, due to the presence of multiple RNA-binding domains, the proteins belonging to the IGF2BP family (IGF2BP1-3) exhibit more complex binding patterns. For example, it has been shown that IGF2BP1 usually interacts with the RNA forming two binding sites, that can be found at varying distances and orientations in functional target sequences (Patel *et al.*, 2012). Since DeepBind models are trained on RNAcompete data, they account for local sequence motifs and their predictive performance might be compromised for RBPs that exhibit two or more disconnected binding sites.

The explanation of the general superior performance of ProtScan might lie in the two main differences with the competitor methods: the consensus voting that produces more robust predictions, and the exploitation of the context information to localize a protein binding site. GraphProt and ProtScan employ a similar approach to generate the RNA features and their models were trained from and tested on the same eCLIP derived datasets, but interaction profiles predicted by ProtScan are more robust due to the consensus voting step. Although the proposed extension of DeepBind computes the average over a sliding window approach to produce per nucleotide predicted interaction profiles, DeepBind directly estimates the interactivity of each RNA window. In this way only interaction information is taken into account. Differently, ProtScan uses the information contained in the RNA window of the context of a binding site to localize its center.

Chapter 6

PTRcombiner

The progress in mapping RNA-protein and RNA-RNA interactions at a transcriptome-wide level allowed to gain valuable information to investigate post-transcriptional gene regulation. Unfortunately, the available data does not reveal RNA molecules that could be targeted by multiple post-transcriptional regulators simultaneously. In this chapter, I present PTRcombiner (Post-Transcriptional Regulation combinatorial miner), an approach to mine the combinatorial nature of post-transcriptional trans-acting factors (RBPs and miRNAs). PTRcombiner is divided into two activity components. The first, "mining combinatorial features" takes as input an interaction map between trans-acting factors (RBPs and miRNAs) and mRNAs, and finds groups of trans-acting factors having in common a conspicuous number of mRNA targets. The second, "analyzing combinatorial features" evaluates the biological characteristics of the clusters identified by the pattern set miner. The identification of clusters of trans-acting factors is performed by factorizing, in a Boolean fashion, the interaction matrix between trans-acting factors and mRNAs. This enables the identification of different, and possibly overlapping, groups (clusters) of trans-acting factors that jointly account for the majority of the interactions in the interaction map. Although Boolean matrix factorization has been employed in data mining to identify pattern sets, its application to computational biology, and especially to RNA-protein interaction data analysis, is completely novel.

6.1 Related work

Many computational techniques have been proposed to investigate the interactions among transcription factors, mRNAs and miRNAs. Several approaches concentrate on the prediction of transcriptional networks by: modeling expression levels of genes in terms of the predicted transcription factors that control their transcription rate (Bailly-Bechet *et al.*, 2010; Asif and Sanguinetti, 2011; Ament *et al.*, 2012); spotting clusters of co-regulated genes (Chesler and Langston, 2007); or, more generally inferring portions of regulatory networks (Li *et al.*, 2008; Karlebach and Shamir, 2008).

Surely, the automated identification of combinatorial patterns at a post-transcriptional level would also be of paramount interest. Some efforts have been spent in analyzing miRNA-mediated interactions, by identifying putative feed-forward loops, where a transcription factor controls the transcription of a miRNA, and together they regulate the translation of a set of target genes (Re *et al.*, 2009; Friard *et al.*, 2010; El Baroudi *et al.*, 2011). More generally, by combining the output of individual miRNA target predictors, PicTar (Krek *et al.*, 2005) infers the combinatorial binding affinity of a set of miRNAs on a target mRNA. Later, ComiR (Coronnello and Benos, 2013) improved the combinatorial model by accounting for miRNA expression levels to rebalance the single prediction scores. Even if limited to miRNA-mRNA interactions, these methods represent the initial attempts to unveil the combinatorial nature of post-transcriptional regulation at a genome-wide scale. However, both approaches expect to specify in advance the set of miRNA to be tested, limiting their applicability to the validation of putative clusters of miRNA regulators, and preventing the efficient discovery of unknown combinatorial patterns, because a comprehensive enumeration of all combinations of miRNAs is computationally infeasible for all but the smallest sets of regulators.

Under the assumption that co-expressed genes are more likely to be functionally related, potential gene networks can be derived from transcriptome expression data. Joshi *et al.* (2011) proposed a probabilistic approach (LeMoNe) that, by accounting for both transcriptional and post-transcriptional regulators, infers module networks in yeast. The resulting putative sets of regulators represent interesting hypotheses of regulatory pathways in specific biological conditions (i.e. stress conditions). However,

the required input for the probabilistic method are explicit translational profiling time series.

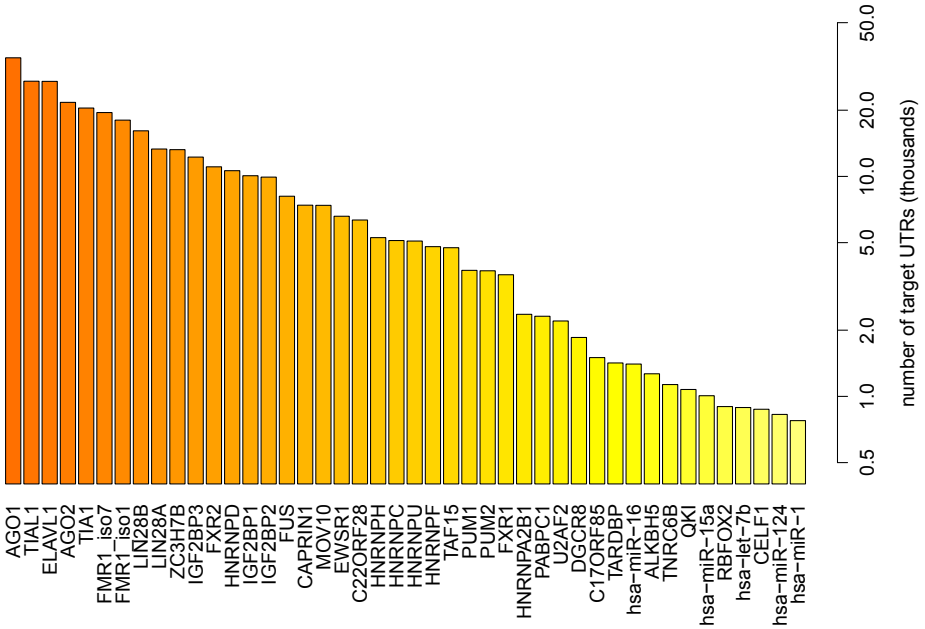
6.2 Materials and methods

In this section, I first describe the data used in our analysis, followed by the formal definition of the computational model used to extract the clusters of trans-acting factors and the explanation of the analysis techniques employed to show the quality of the extracted clusters.

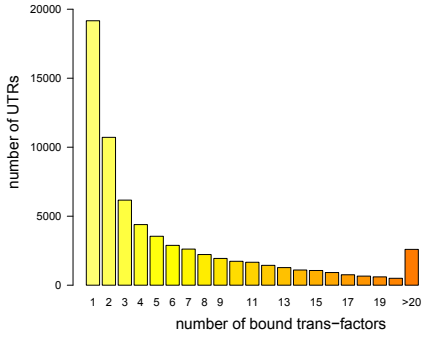
6.2.1 Dataset

The AURA 2 database (July 2013) (Dassi *et al.*, 2014) is a manually curated and comprehensive catalog of mRNA untranslated regions (UTRs) and their regulatory annotations, including interactions with trans-acting factors (mainly RBPs and miRNAs). The annotations come from a wide range of experimental techniques, including CLIP, RIP, SELEX, and RNAcompete. A subset of these techniques, represented by CLIP experiments, allows to pinpoint RNA-protein interactions and to obtain the positional information about the region of the RNA that is bound by the RBP, while the other methods, are only able to detect the presence of an interaction between a transcript and a trans-acting factor without the positional information of the specific binding site.

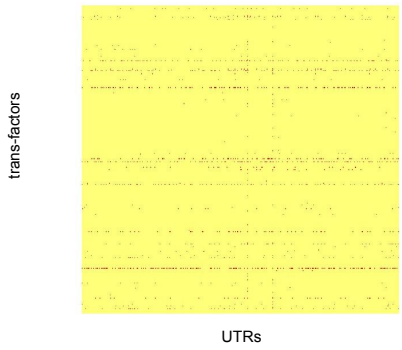
I considered the entire set of human interactions annotated in AURA 2, considering both RBPs and miRNAs as trans-acting factors. The number of UTRs bound by each trans-acting factor varies from 1 to 34,616, with median of 13 and a mean value of 695. Figure 6.1a displays a histogram of the number of UTR targets bound by the most interacting trans-acting factors. On the other hand, the number of trans-acting factors bound to the same UTR ranges from 1 to 64, with median 3 and mean 6 (the distribution is shown in Figure 6.1b). The interaction information contained in the AURA 2 human dataset was then encoded into a Boolean matrix Y with 67,962 rows corresponding to the number of human UTRs with at least one annotated interaction, and 569 columns corresponding to the number of trans-acting factors, RBPs and miRNAs, present in the dataset. Figure 6.1c represents the interaction matrix, where Y_{ij} is equal to 1 if trans-acting factor j interacts with UTR i, and 0 otherwise. Collectively, the selected annotated



(a) Number of target UTRs per trans-acting factor. Human trans-acting factors (RBPs and miRNAs) are ordered according to the number of annotated target UTRs. Trans-acting factors with less than 750 distinct UTR targets are not shown.



(b) Distribution of the number of distinct trans-acting factors bound to the same UTR.



(c) Graphical representation of the Boolean interaction matrix, derived from the input pairwise interactions. Each row corresponds to a trans-acting factor, each column to a UTR. Positive interactions are displayed in red.

Figure 6.1: Interactions annotated in AURA 2 (July 2013).

interactions are 395,395, resulting in a density of the interaction matrix of 0.01.

6.2.2 Boolean matrix factorization

The main focus of PTRcombiner is to discover clusters of co-acting trans-acting factors. This can be accomplished by factorizing a $n \times m$ Boolean matrix Y, representing the interaction maps in the available dataset, into two Boolean matrices representing the basis decomposition of the matrix Y. For example, in the AURA 2 human dataset, the interaction matrix Y contains interactions between m trans-acting factors (either RBPs or miRNAs) and n UTRs.

The "mining combinatorial features" module employs the algorithm for Boolean matrix factorization, originally presented in Miettinen *et al.* (2008), for identifying clusters of trans-acting factors that bind the same set of targets. Let Y be a $n \times m$ Boolean matrix which represents trans-acting factor–target interactions, where m is the number of trans-acting factors, and n the number of targets. The rows of the matrix represent the observations, i.e. the targets, while the columns represent the attributes, i.e. the trans-acting factors. A basis vector identifies a set of correlated attributes or, in other words, a cluster of co-acting trans-acting factors.

Let U and C be binary matrices of size $n \times k$ and $m \times k$, respectively. The $n \times m$ matrix $U \circ C$ represents the Boolean product between U and C. More intuitively, C is the cluster matrix that states the cluster composition (in terms of trans-acting factors), and U is the usage matrix that shows how clusters of trans-acting factors interact with single targets.

Given a binary $n \times m$ interaction matrix Y and a positive integer $k \leq \min\{n, m\}$, the aim is to find two Boolean matrices $U \in n \times k$ and a $C \in m \times k$ that minimize

$$|Y - U \circ C^\top| = \sum_{i=1}^{n} \sum_{j=1}^{m} |Y_{ij} - (U \circ C^\top)_{ij}| \tag{6.1}$$

Finding an exact solution to Equation 6.1 is a $\mathcal{NP}$-hard problem, that requires non-polynomial time to be solved exactly. For this reason, PTRcombiner uses an approach that finds an approximate solution to the factorization problem. The solving technique, originally proposed in Miettinen

et al. (2008) populates the cluster matrix C, and accordingly the usage matrix U by trying to cover the interactions in the matrix Y, in a greedy manner. The greedy approach prioritizes the covering of denser rows of the interaction matrix, i.e. with a high proportion of ones. First, a pool of candidate basis vectors is computed from the association scores between pairs of trans-acting factors. Then, k basis vectors are selected in a greedy fashion. Let A' be a $m \times m$ matrix that contains the association scores between couples of trans-acting factors. $A \in m \times m$ is defined as the Boolean matrix of the candidate basis vectors, where $A_{ij} = 1$ if the association score between trans-acting factor i and trans-acting factor j is grater than a certain threshold $\tau \leq 1$, and 0 otherwise.

PTRcombiner can utilize two approaches to estimate the association score between trans-acting factors. The two association scores have different characteristics, that promote the the discovery of different types of clusters of co-acting trans-acting factors. The standard version, presented in Miettinen *et al.* (2008) uses an unbalanced association score, where the association of the i-th trans-acting factor with the j-th one is computed as $y(i \Rightarrow j) = \langle \mathbf{y}_{.i}, \mathbf{y}_{.j} \rangle / \langle \mathbf{y}_{.i}, \mathbf{y}_{.i} \rangle$ ($\langle \cdot, \cdot \rangle$ is the inner product between vectors). In general $y(i \Rightarrow j) \neq y(j \Rightarrow i)$, resulting in an asymmetric association matrix. The i-th row of A, that represents the i-th candidate basis vector (cluster) is computed using the i-th trans-acting factor as seed: $A_{ij} = 1$ if the percentage of shared targets between the i-th and the j-th trans-acting factors is at least τ times the number of targets of the i-th trans-acting factor, and 0 otherwise. This association score is only normalized with respect to the number of targets of the seed trans-acting factor. By consequence, trans-acting factors with many targets are prone to have a high association scores with most of the trans-acting factors with only few interactions, and thus to appear in multiple clusters. This association score fosters the identification of combinatorial interactions between trans-acting factors with heterogeneous degrees of specificity (e.g. RBPs and miRNAs). On the other hand, clusters formed by only specific trans-acting factors tend to be discarded by the greedy procedure.

In order to address this bias of the greedy technique, another association score is proposed. The balanced association score is based on the vector cosine similarity: $y(i \Leftrightarrow j) = \langle \mathbf{y}_{.i}, \mathbf{y}_{.j} \rangle / \sqrt{\langle \mathbf{y}_{.i}, \mathbf{y}_{.i} \rangle \cdot \langle \mathbf{y}_{.j}, \mathbf{y}_{.j} \rangle}$. The resulting association matrix is symmetric and it promotes the identification of clusters

with higher homogeneity in terms of number of targets of their trans-acting factors.

6.2.3 Biological characterization

After finding the clusters of trans-acting factors, the "analyzing combinatorial features" module allows to characterize the mined clusters. It analyzes the RNA targets associated to a cluster of trans-acting factors, i.e. the targets bound by all the trans-acting factors in the cluster.

6.2.3.1 Target overlap

The overlap between the targets of two different clusters is computed using the Jaccard similarity. It is defined as the ratio between the size of the intersection and the size of the union of two sets. This similarity measure ranges from 0, when the two sets do not share any element, to 1, when the two sets contain the same elements.

6.2.3.2 Functional analysis

In order to individuate the functional enrichments of a set of targets bound by a cluster of trans-acting factors, Gene Ontology enrichment analysis is performed with the topGO package[1], using the Fisher's exact test statistics and the "elim" method for dealing with the GO graph structure, that prefers more specialized nodes of the ontology. A p-value threshold of 0.05 is used to determine the significance of over-representation. The enrichment analysis is performed on the list of genes regulated by each cluster of trans-acting factors. In order to compare the functional enrichments associated with targets of single trans-acting factors with enrichments associated to targets of clusters, the enrichment analysis is also performed on the list of genes interacting with each single trans-acting factor of a cluster.

In addition, the semantic similarity between two lists of enriched GO terms is computed using the GOsemsim package (Yu *et al.*, 2010), with Wang's method to determine pairwise semantic similarities between GO terms and the BMA (best-match average) method to combine the semantic similarity scores of multiple GO terms.

[1] http://www.bioconductor.org/packages/2.13/bioc/html/topGO.html

6.2.4 RBP-binding site classifier

When positional interaction information is available, the "analyzing combinatorial features" module permits the classification of the RNA binding sites of the trans-acting factors in a cluster, allowing to determine whether multiple RBPs exhibit the same RNA site affinity or not.

Information resulting from experimental techniques, is often corrupted (to some extent) from different noise sources. The most relevant source of noise is represented by the false negatives. A fraction of binding sites might remain undetected due to the intrinsic dependency on cell lines, tissues or environmental conditions in which the experiment was performed. Additionally, the post-processing analysis, that include mapping and peak detection, might increase the number of false negatives due to the burden of dealing with splice junctions and the stringent thresholds required for a confident detection. Therefore, computational approaches for RBP target site modeling are helpful assets for dealing with the low signal-to-noise ratio of the available experimental techniques. To establish whether RBPs are likely to interact with the same RNA sites or not, first *in silico* models of the preferred target sites of the different RBPs in a cluster are built, and then a machine learning algorithm to discriminate between binding sites of two different RBPs is trained, for all possible pairwise combinations. When the algorithm confidently distinguishes between their binding sites, the two RBPs are likely to have different binding affinities. On the other hand, the incapability of performing the discrimination task points at the hypothesis of analogous binding sites. The discrimination task is based on a kernel machine binary classifier able to work with RNA sequences, and to compute the similarity between base sequences in terms of their predicted secondary structures. Since RNA-protein interactions are not solely driven by sequence specificities, the use of structural components in this discrimination task yields a strong biological significance.

Kernelized machine learning approaches embed a suitable similarity function, called kernel, that enables to perform learning tasks over arbitrary data structures, like graphs. This allows the modeling of RNA secondary structures in a natural way: with vertices representing nucleotides and edges representing the nucleotide bonds, i.e. backbone phosphate bonds and base-pairing bonds. As graph kernel the NSPDK (Costa and De Grave, 2010) is employed. It generalizes the concept of (gapped) k-mers string kernels

to graphs. Instead of measuring the fraction of common small contiguous subsequences (k-mers) between two strings, NSPDK determines the similarity between two graphs by counting the shared fraction of neighborhood subgraphs. A neighborhood subgraph is induced by all vertices within a specified radius from a given root vertex, where the distance between two vertices is the length of the shortest path between the vertices. Clearly establishing graph homomorphism is harder than spotting two equivalent strings. In Costa and De Grave (2010) an efficient approximation based on hashing a quasi-canonical graph representation is used.

In Heyne *et al.* (2012) NSPDK was applied to a represent graphs of RNA folding structures. The leading idea was to rely not only on the RNA minimum free energy configuration, that is commonly error prone, but to benefit from efficient dynamic programming algorithms (Giegerich *et al.*, 2004) to sample multiple putative secondary structures for the given sequence. These multiple secondary structures consider a small number of representatives that are both structurally diverse and energetically stable. All the folding hypotheses of an RNA are considered simultaneously in a comprehensive disconnected graph.

The binding site classification is accomplished merging all these ideas in a unified framework. Given an RNA region: first, a sample of stable and diverse folding structures is computed and encoded in a disconnected graph; second, the graph is turned into a feature representation by the NSPDK; and finally, feature representations of the binding sites of different RBPs are discriminated with an SVM.

6.3 Results and discussion

In this section I present the experimental results obtained running PTRcombiner on the AURA 2 dataset. First, the clusters of trans-acting factors found in the AURA 2 dataset are displayed and analyzed. Then, an example of the usage of the RBP-binding site classifier is shown. Finally, another set of clusters is extracted from the AURA 2 database by exploiting the cosine similarity based association score instead of the unbalanced association score proposed in Miettinen *et al.* (2008).

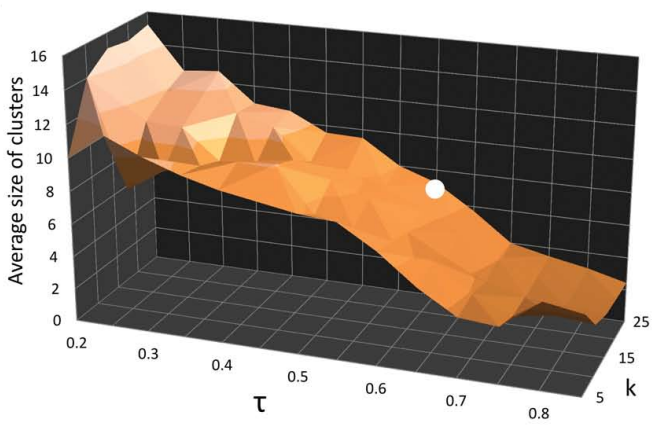


Figure 6.2: Exploration of the hyperparameter space. The average size (i.e. the number of trans-acting factors members) of the identified clusters is displayed at different combinations of k and τ values. The white dot marks the configuration of the hyperparameters selected to extract the clusters in the presence of recurrent trans-acting factors.

6.3.1 Mining combinatorial features

The key focus of PTRcombiner is to identify clusters of trans-acting factors (RBPs and/or miRNAs) that bind the same set of RNA targets. The aim was to employ Boolean matrix factorization (Section 6.2.2) on the AURA 2 human dataset (Section 6.2.1) to identify multiple overlapping clusters, that jointly represent most of the known interactions between trans-acting factors and UTRs. In this section I show how PTRcombiner was used on the AURA 2 dataset to extract clusters of trans-acting factors.

The greedy approach to Boolean matrix factorization has two hyperparameters: the number of clusters to return (k), and a threshold (τ) that controls the amount of shared targets inside clusters. The higher the threshold, the more targets should be shared among the trans-acting factors in order to form a cluster. The algorithm returns a ranked list of clusters, sorted by coverage, i.e. number of targets of the cluster.

In order to select the hyperparameter values, the average cluster size (number of trans-acting factors) was analyzed while varying k and τ values (Figure 6.2). I noted that, once τ was fixed, the value of k did not affect the average cluster size, that seemed to rely only on the value of τ. The τ value was chosen according to the average cluster size of the retrieved

clusters. The τ value that produced an average cluster size as close as possible to the average number of trans-acting factors bound to a single UTR was selected. The selected value was $\tau = 0.6$, resulting in clusters composed of averagely 6 trans-acting factors. Since the k value did not affect, at least in the considered hyperparameter space, the average cluster size, the selected value was $k = 25$. Table 6.1 shows the clusters found by PTRcombiner with the selected hyperparameters. The top nine clusters are composed only by RBPs, as well as the clusters R11 to R19, R22 and R25. The first cluster displaying co-occurrence of RBPs and miRNAs is R10, followed by clusters R20, R21, R23 and R24. No clusters composed uniquely of miRNAs are present in the list. Moreover, 5 out of 25 clusters do not represent real combinations, as they are singletons composed of only one trans-acting factor. Since, the algorithm is guided by the coverage of the interaction matrix, a singleton cluster is extracted whenever the trans-acting factor has a significant number of interactions, and those interactions are not in common with any other trans-acting factor in the dataset.

As reported in Section 6.2.1, trans-acting factors have a quite different number of UTR targets. The greedy approach for Boolean matrix factorization is driven by coverage, and therefore it is inherently biased towards the selection of clusters composed of widely interacting trans-acting factors. By analyzing the composition of the clusters reported in Table 6.1, I observed that some trans-acting factors were almost ubiquitously present in the clusters, e.g. the Argonaute proteins AGO1 and AGO2, and the well-known RBP ELAVL1/HuR, occur in 19, 15 and 17 out of 25 clusters, respectively. AGO1 and AGO2 are components of the RNA-induced silencing complex (RISC), the protein complex responsible for mRNAs down-regulation (Pasquinelli, 2012). By binding different classes of small ncRNAs, such as miRNAs and small interfering RNAs (siRNAs), these proteins bind mRNA through sequence complementarity, and performing the silencing of the bound targets. Given the widespread activity of AGO1 and AGO2, it was not surprising to find them in almost all the clusters. As depicted by the results, AGO1 and AGO2 have also been found to interact with ELAVL1/HuR (Landthaler *et al.*, 2008). The Argonaute proteins and ELAVL1/HuR exhibit different mRNA binding affinity: AGO proteins usually bind the edges of the UTRs, while ELAVL1/HuR binds uniformly along UTRs, with vanishing activity in proximity of the stop codon and the

Table 6.1: List of the inferred clusters in the presence of recurrent trans-acting factors.

Class	Cluster	trans-acting factors
RBP	Clust R01	AGO1, AGO2, ELAVL1, FMR1_iso1, FMR1_iso7, FXR2, LIN28A, LIN28B, MOV10, TIA1, TIAL1, ZC3H7B
RBP	Clust R02	AGO1, AGO2, ELAVL1, IGF2BP1, IGF2BP2, IGF2BP3, TIAL1
Singleton	Clust R03	AGO1
RBP	Clust R04	ELAVL1, HNRNPD
RBP	Clust R05	AGO1, AGO2, ELAVL1, EWSR1, FMR1_iso1, FUS, LIN28A, LIN28B, TAF15, TIA1, TIAL1, ZC3H7B
RBP	Clust R06	AGO1, ELAVL1, TIA1, TIAL1
RBP	Clust R07	AGO1, FMR1_iso1, FMR1_iso7
RBP	Clust R08	AGO1, AGO2, CAPRIN1, ELAVL1, FMR1_iso1, FMR1_iso7, LIN28B, TIA1, TIAL1, ZC3H7B
RBP	Clust R09	AGO1, AGO2, C22ORF28, ELAVL1, FMR1_iso1, FMR1_iso7, LIN28B, TIA1, TIAL1, ZC3H7B
RBP-miRNA	Clust R10	LIN28A, LIN28B, hsa-miR-221*
RBP	Clust R11	AGO1, HNRNPH
RBP	Clust R12	AGO1, AGO2, ELAVL1, FMR1_iso1, HNRNPC, TIA1, TIAL1
Singleton	Clust R13	PUM1
RBP	Clust R14	AGO1, AGO2, ELAVL1, FMR1_iso1, FMR1_iso7, HNRNPU, TIA1, TIAL1
RBP	Clust R15	AGO1, AGO2, ELAVL1, FMR1_iso1, FMR1_iso7, HNRNPF, TIA1, TIAL1
RBP	Clust R16	AGO1, AGO2, ELAVL1, EWSR1, FMR1_iso1, FMR1_iso7, FXR1, FXR2, LIN28A, LIN28B, TIA1, TIAL1, ZC3H7B
RBP	Clust R17	AGO1, AGO2, ELAVL1, FMR1_iso1, IGF2BP1, IGF2BP2, IGF2BP3, PUM2, TIA1, TIAL1
Singleton	Clust R18	PABPC1
Singleton	Clust R19	U2AF2
RBP-miRNA	Clust R20	AGO1, AGO2, ELAVL1, FMR1_iso1, IGF2BP1, IGF2BP2, IGF2BP3, TIA1, TIAL1, hsa-miR-130a, hsa-miR-130b, hsa-miR-148a, hsa-miR-148b, hsa-miR-301a, hsa-miR-301b
RBP-miRNA	Clust R21	AGO1, AGO2, ELAVL1, FMR1_iso1, IGF2BP1, IGF2BP2, IGF2BP3, TIA1, TIAL1, hsa-miR-15a, hsa-miR-15b, hsa-miR-16, hsa-miR-424
Singleton	Clust R22	DGCR8
RBP-miRNA	Clust R23	AGO1, AGO2, ELAVL1, FMR1_iso1, IGF2BP1, IGF2BP2, IGF2BP3, TIA1, TIAL1, hsa-miR-106b, hsa-miR-17, hsa-miR-20a, hsa-miR-320, hsa-miR-93
RBP-miRNA	Clust R24	AGO1, AGO2, ELAVL1, IGF2BP1, IGF2BP2, IGF2BP3, TIAL1, hsa-let-7a, hsa-let-7b, hsa-let-7c, hsa-let-7d, hsa-let-7e, hsa-let-7f, hsa-let-7g, hsa-let-7i
RBP	Clust R25	AGO1, AGO2, ELAVL1, FMR1_iso1, HNRNPA2B1, TIA1, TIAL1

polyadenylation site (Lebedeva *et al.*, 2011). ELAVL1/HuR is known to be mostly expressed in tissues and to bind AU-rich elements in the 3' UTRs of numerous mRNAs (Lebedeva *et al.*, 2011; Mukherjee *et al.*, 2011). It has also been shown that ELAVL1/HuR displays competitive and cooperative interactions with miRNAs/RISC (Kim *et al.*, 2009), and that it is part of a complex mRNA network that coordinates gene expression (Simone and

Keene, 2013). These findings support the theory that trans-acting factors frequently occurring in the clusters have the highest number of interactions. These trans-acting factors are called "recurrent" and the respective clusters are identified by Ri, where R stands for recurrent and i represent the cluster number (ranging from 1 to 25).

I was also interested in spotting clusters composed of trans-acting factors with a narrower spectra of interactions and therefore less likely to occur in the clusters. For this reason, I removed all trans-acting factors that appeared in more than one cluster in Table 6.1, and ran another iteration of the mining procedure. This second iteration focused on trans-acting factors that appeared in maximum one of the clusters in Table 6.1, named sporadic trans-acting factors. Similarly to the recurrent case, I analyzed the average cluster size while varying the hyperparameters k and τ. This time, the optimal choice of τ was 0.4, returning clusters composed of averagely 3 trans-acting factors. This number corresponds to the average number of sporadic trans-acting factors bound to each UTR. Sporadic clusters are displayed in Table 6.2 and they are identified by Si, where S stands for sporadic and i represents the cluster number (ranging from 1 to 25). The majority of clusters (15 out of 25) are singletons. In contrast with the results obtained when recurrent factors were included, here I observed that 4 clusters are composed exclusively of miRNAs (S09, S14, S16 and S22). Another alluring comment regards PUM2, that was found as a member of the recurrent cluster R17, while here it is present in two distinct clusters with different sets of miRNAs (S10 and S21). PUM2 is known to act as a translational repressor in several organisms, being involved in dendritic RNA localization and silencing (Vessey *et al.*, 2006) and regulating synaptic formation (Vessey *et al.*, 2010). In accordance with this result, a pervasive interaction between Pumilio proteins and the miRNA regulatory system has been suggested (Galgano *et al.*, 2008), indicating that, in translational regulation, the synergy between RBPs and miRNAs may be more usual than previously thought. Recently, a computational analysis suggested that the binding sites of particular sets of miRNA localize within 50 nucleotides from PUM2 binding sites (Jiang *et al.*, 2013), supporting the cooperative hypothesis between PUM2 and miRNA in mRNA degradation. These discoveries support the clusters where PUM2 acts in combination with different miRNAs, especially regarding hsa-miR-221 and hsa-miR-222, that are in cluster

Table 6.2: List of the inferred clusters composed of sporadic trans-acting factors.

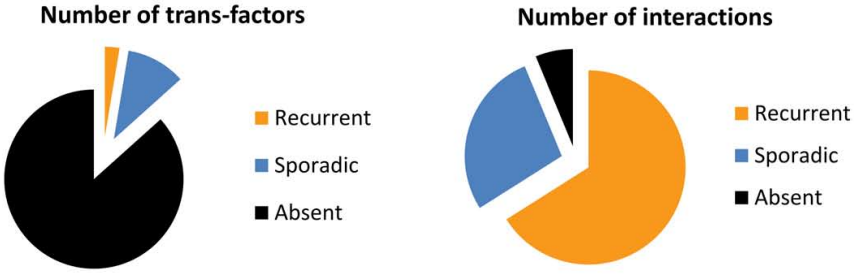
Class	Cluster	trans-acting factors
Singleton	Clust S01	HNRNPD
RBP	Clust S02	CAPRIN1, FUS, FXR1, MOV10, TAF15
Singleton	Clust S03	HNRNPH
RBP	Clust S04	C22ORF28, CAPRIN1, MOV10
Singleton	Clust S05	HNRNPC
Singleton	Clust S06	HNRNPU
Singleton	Clust S07	HNRNPF
Singleton	Clust S08	PUM1
miRNA	Clust S09	hsa-miR-15a, hsa-miR-15b, hsa-miR-16, hsa-miR-424
RBP-miRNA	Clust S10	PUM2, hsa-miR-130a, hsa-miR-130b, hsa-miR-148a, hsa-miR-148b, hsa-miR-19a, hsa-miR-19b, hsa-miR-301a, hsa-miR-301b
Singleton	Clust S11	HNRNPA2B1
Singleton	Clust S12	PABPC1
Singleton	Clust S13	U2AF2
miRNA	Clust S14	hsa-miR-106b, hsa-miR-17, hsa-miR-20a, hsa-miR-93
RBP	Clust S15	MOV10, PUM2
miRNA	Clust S16	hsa-let-7a, hsa-let-7b, hsa-let-7c, hsa-let-7d, hsa-let-7e, hsa-let-7f, hsa-let-7g, hsa-let-7i
Singleton	Clust S17	DGCR8
Singleton	Clust S18	C17ORF85
Singleton	Clust S19	TARDBP
RBP	Clust S20	FUS, MOV10, TAF15
RBP-miRNA	Clust S21	PUM2, hsa-miR-103, hsa-miR-107, hsa-miR-183, hsa-miR-221, hsa-miR-222, hsa-miR-23b, hsa-miR-25, hsa-miR-27a, hsa-miR-27b, hsa-miR-32, hsa-miR-92a, hsa-miR-96
miRNA	Clust S22	hsa-miR-103, hsa-miR-107, hsa-miR-15a, hsa-miR-15b, hsa-miR-16, hsa-miR-29a, hsa-miR-29b, hsa-miR-29c, hsa-miR-424
Singleton	Clust S23	CELF1
Singleton	Clust S24	hsa-miR-124
Singleton	Clust S25	hsa-miR-1

S21 together with PUM2, and seem to conjugate with the RBP (Jiang *et al.*, 2013).

Even though a small fraction of the trans-acting factors is present in the clusters (Figure 6.3a), the majority of the known interactions are covered by the identified clusters (Figure 6.3b). The majority of the trans-acting factors are not retained in any of the clusters because of the lack of available information. Given the novelty of the experimental techniques, it is obvious that more information is required to exhaustively enumerate the combinatorial features of the human post-transcriptional regulation.

6.3.2 Biological characterization

After finding the clusters of trans-acting factors, I characterized them under a biological point of view. I evaluated their RNA targets and the respective overlap, the enriched ontological terms and the similarity among the en-



(a) Proportion among the number of recurrent, sporadic, and absent trans-acting factors.

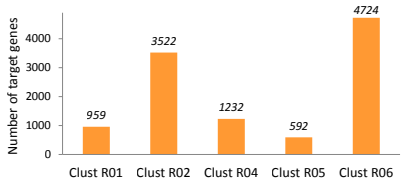
(b) Proportion among the number of interactions associated with recurrent, sporadic, and absent trans-acting factors.

Figure 6.3: Analysis of the recurrent, sporadic and absent trans-acting factors.

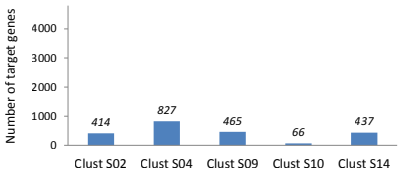
riched terms. This characterization was performed only on the non-singleton clusters.

Given the high number of RNA targets of recurrent trans-acting factors, several hundred genes are co-regulated by recurrent trans-acting factors (Figure 6.4a). On average, 2,206 genes are regulated by the first five clusters (excluding singletons), ranging from 592 of cluster R05 to 4,724 of cluster R06. Given their greater specificity, the number of target genes is significantly lower when taking into account the clusters of sporadic trans-acting factors, (Figure 6.4b). The first five non-singleton sporadic clusters regulate averagely 442 genes, ranging from 66 of cluster S10 to 827 of cluster S04.

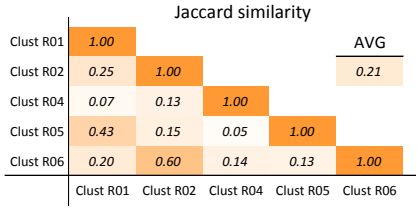
Each cluster of trans-acting factors regulates a specific set of genes, here I accounted for the overlap among target genes of different clusters. When considering clusters obtained in presence of recurrent trans-acting factors, the average overlap is 21% (Figure 6.4c), suggesting that PTRcombiner was able to spot clusters of trans-acting factors that target different sets of genes. However, in some cases the percentage of shared targets is higher, e.g. cluster R01 and cluster R05 share 43% of their targets. This phenomenon is due to the high overlap of trans-acting factors between the two clusters, and it can be also observed when considering clusters R02 and R06, that share 60% of their targets. In this extreme case, cluster R06 shares with R02 almost all the trans-acting factors (AGO1, ELAVL1, and TIAL1). Considering the clusters of sporadic elements the average overlap decreases to 7%, 3 times



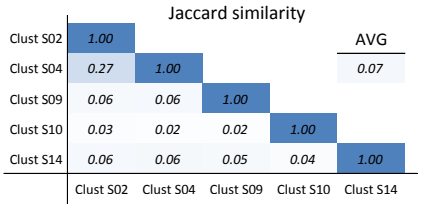
(a) Number of genes targeted by the top five ranking clusters that include recurrent trans-acting factors.



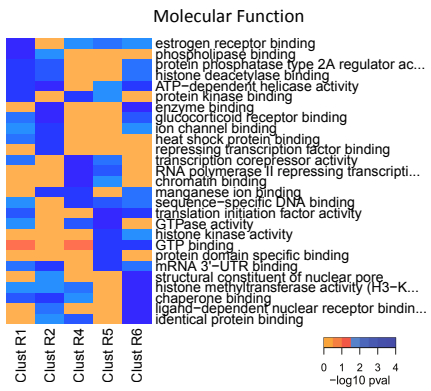
(b) Number of genes targeted by the top five ranking clusters of sporadic trans-acting factors.



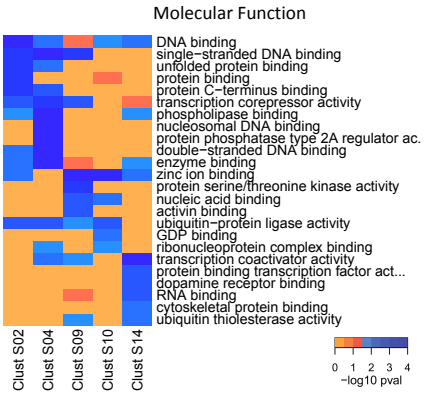
(c) Jaccard similarities among the top five ranking clusters that include recurrent trans-acting factors.



(d) Jaccard similarities among the top five ranking clusters of sporadic trans-acting factors.



(e) Heatmap showing the top enriched Molecular Function GO terms associated with genes targeted by the top five ranking clusters obtained including recurrent trans-acting factors.



(f) Heatmap showing the top enriched Molecular Function GO terms associated with genes targeted by the top five ranking clusters obtained from sporadic trans-acting factors.

Figure 6.4: Biological characterization of the recurrent and sporadic clusters.

less with respect to the previous analysis (Figure 6.4d). Here, the higher overlap is registered between clusters S02 and S04, that have 27% of common RNA targets. This reduced overlap supports the efficacy of repeating the individuation of clusters considering only sporadic trans-acting factors, that allowed to find small-sized sets of genes regulated by trans-acting factors with a low number of annotated interactions.

In order to address the biological relevance of the mined clusters, I also performed Gene Ontology enrichments analysis. The aim is to identify common and biologically coordinated mechanisms or processes that administer cellular outcomes. This analysis accounts for general biological annotations allowing to compare the clusters by the gene ontology (GO) enrichment of their target RNAs. Figure 6.4e and 6.4f show the enrichments of the top enriched GO terms for each cluster. The modularity of the enriched terms scattered along the columns of the heatmap clearly indicates a high level of diversification of molecular functions carried by the sets of genes regulated by the different clusters. The only visible exception is represented by clusters S02 and S04, that display very close enrichment signatures, mirroring the strong similarity, in term of trans-acting factors, observed between the two clusters.

Finally, I assessed the change in ontological enrichment between the gene targets of all trans-acting factors belonging to a cluster and the gene targets of the single trans-acting factors. This intra-cluster comparison enabled the potential identification of emerging features, that are exclusively associated to the entire cluster and not to specific trans-acting factors forming a cluster. Figure 6.5 shows an example of this analysis performed on cluster S02. The target genes associated to the cluster exhibit specific enrichments that are not associated to any of the RBPs forming the cluster: "cell division" in biological process (BP), "nuclear speck" in cellular component (CC), and "transcription corepressor activity" in molecular function (MF). These results suggest that clusters of trans-acting factors have emergent and specific combinatorial properties that are not usually exerted by their components alone.

6.3.3 RBP-binding site classification

In order to give deeper insight information about the mined clusters of trans-acting factors, in this section I show that the classification method can detail

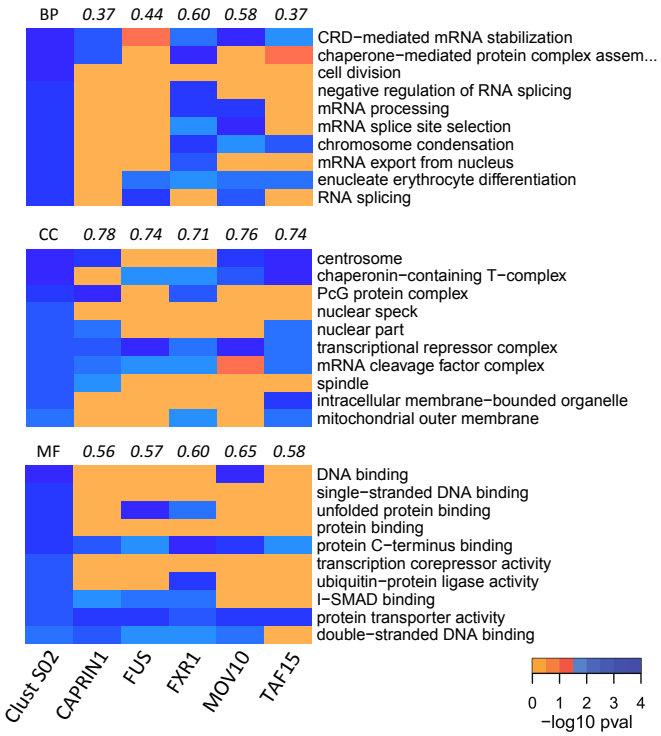
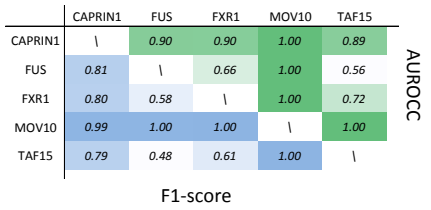


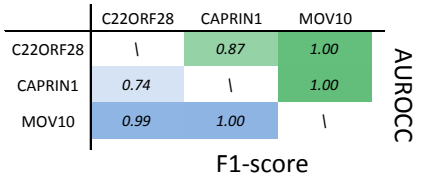
Figure 6.5: Intra-cluster GO enrichment analysis of cluster S02. Comparison of the ontological enrichment between the gene targets of all trans-acting factors belonging to the cluster and the gene targets of the single trans-acting factors. The comparison is shown for all three Gene Ontologies: Biological Process (BP), Cellular Component (CC), and Molecular Function (MF). The top rows of each panel report the semantic similarity between the enriched terms associated to single trans-acting factors and the ones associated to the cluster.

the binding affinities of RBPs in a cluster. The basic idea is that whenever two RBPs, belonging to the same cluster and therefore co-interacting with the same set of RNAs, are characterized by similar binding site affinity, then a concurrent binding, either competitive or cooperative, might occur. This type of analysis was limited to clusters formed by only RBPs with positional interaction information (e.g. CLIP-seq).

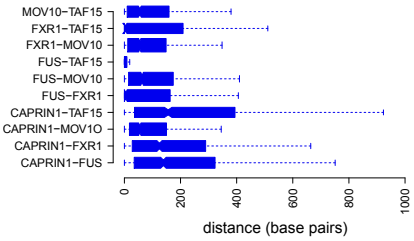
As an example, I analyzed the first two non-singleton clusters of sporadic trans-acting factors, i.e. cluster S02 composed by CAPRIN1, FUS, FXR1, MOV10, and TAF15, and cluster S04 formed by C22ORF28, CAPRIN1, and MOV10. For each RBP, I randomly selected 2,500 RNA stretches (of 20–70 nt) from the available binding coordinates annotated in the AURA 2



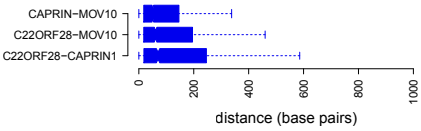
(a) Pairwise classification performance values for cluster S02.



(b) Pairwise classification performance values for cluster S04.



(c) Distributions of the pairwise distances between binding sites of trans-acting factors belonging to cluster S02.



(d) Distributions of the pairwise distances between binding sites of trans-acting factors belonging to cluster S04.

Figure 6.6: RBP site classification on cluster S02 and S04.

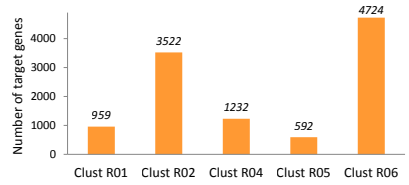
database. The classification performance for clusters S02 and S04 is shown in Figure 6.6a and 6.6b, respectively. Performance was evaluated according to the AUROCC and F1-score measures. AUROCC evaluates the quality of a classifier while varying the threshold to decide whether a prediction should be considered positive or not. An AUROCC value of 0.5 corresponds to the one of the random predictor, while an AUROCC value of 1 indicates perfect discrimination. The F1-score is the harmonic mean between precision and sensitivity, that trades off the two complementary measures. Analyzing the cluster S02, the classifier was able to discriminate the binding sites of only a subset of the RBPs in the cluster. Very good performance can be observed for CAPRIN1 (with an average AUROCC of 0.92 and an average F1 of 0.85) and MOV10 (with an average AUROCC and F1 of 1.0). On the other hand, FUS and TAF15 seem to have more similar binding sites. In fact, an AUROCC of 0.56 suggests that these proteins share similar if not identical binding sites. Under a biological point of view, FUS and TAF15 are known paralogues, that belong to the FET family of RNA-binding proteins (Andersson *et al.*, 2008). The classification scores for cluster S04 are generally high, suggesting that the UTR stretches that are bound by the

three proteins in the cluster (C22ORF28, CAPRIN1 and MOV10) are different. Figures 6.6c and 6.6d show the distribution of the pairwise distances between binding sites of couples of RBPs. Clearly, FUS and TAF15 have much closer binding sites with respect to all the other couples of RBPs. Also the distance between binding sites of FXR1 and FUS or TAF15 is low, but still not comparable with the one between the two parologue proteins FUS and TAF15. A large average distance can be observed for all the other cases, confirming the good classification scores obtained with the classifier.

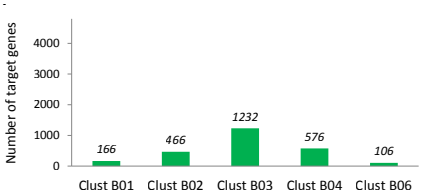
These use cases demonstrated how the *in silico* modeling of RNA-proteins interactions can help the investigation of RBPs combinatorial effects. This modeling approach is more resilient to noisy experimental data, since it can recover missed interactions (false negatives). Predictive models allow more sophisticated investigations with respect to simple analysis of the experimental evidence. For instance, a competitive effect can be hypothesized when two RBPs exhibit a compatible binding preference, even if the experimental data do not report overlapping interaction areas. Conversely, if the model predicts the target regions to be sufficiently close but not overlapping, a cooperative effect can be hypothesized, even if the experimental data cannot resolve the distinct areas and these are therefore interpreted as overlapping.

6.3.4 Balancing the trans-acting factor sample size

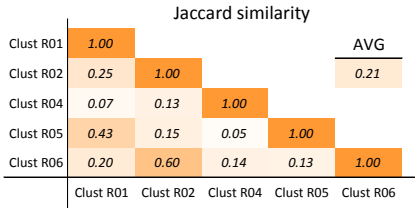
In Section 6.3.1 a bias of the algorithm towards "widely interacting" trans-acting factors was individuated. In Section 6.2.2 an alternative balanced association score to create the pool of possible clusters was described. The original greedy procedure to solve Boolean matrix factorization (Miettinen *et al.*, 2008) constructs a pool of putative clusters by using each candidate trans-acting factor as seed to compute its association score with other trans-acting factors, where the association score is given by the number of shared targets between the two trans-acting factors, normalized by the number of targets of the seed. By definition, this unbalanced score is asymmetric and favors the association of trans-acting factors with few interactions (that are acting as seeds) with those with many interactions (e.g. AGO1) that will easily share a significant fraction of targets with them. The proposed alternative version of the association score uses cosine normalization thereby producing a symmetric association score that weakens the presence of widely interacting trans-acting factors in the majority of clusters.



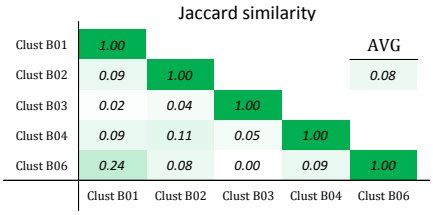
(a) Number of genes targeted by the top five ranking clusters obtained with the unbalanced association score (copy of Figure 6.4a).



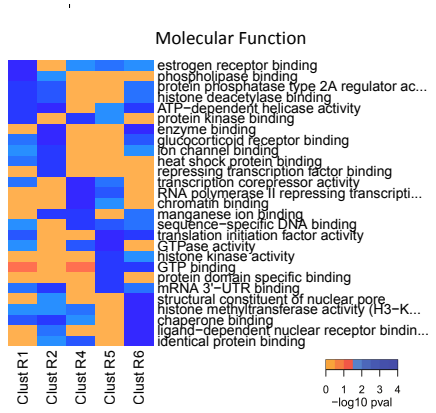
(b) Number of genes targeted by the top five ranking clusters obtained with the balanced association score.



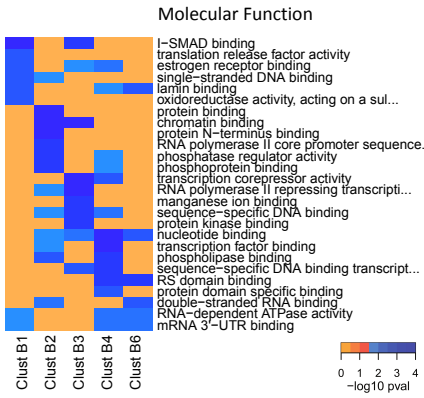
(c) Jaccard similarities among the top five ranking clusters obtained with the unbalanced association score (copy of Figure 6.4c).



(d) Jaccard similarities among the top five ranking clusters obtained with the balanced association score.



(e) Heatmap showing the top enriched Molecular Function GO terms associated with genes targeted by the top five ranking clusters obtained with the unbalanced association score (copy of Figure 6.4e).



(f) Heatmap showing the top enriched Molecular Function GO terms associated with genes targeted by the top five ranking clusters obtained with the balanced association score.

Figure 6.7: Biological characterization of the clusters obtained with unbalanced and balanced association scores.

Table 6.3: List of the inferred clusters using the balanced association score.

Class	Cluster	trans-acting factors
RBP	Clust B01	AGO1, AGO2, CAPRIN1, ELAVL1, EWSR1, FMR1_iso1, FMR1_iso7, FUS, FXR2, HNRNPC, LIN28A, LIN28B, MOV10, TAF15, TIA1, TIAL1, ZC3H7B
RBP	Clust B02	IGF2BP1, IGF2BP2, IGF2BP3, PUM2, TNRC6B
RBP	Clust B03	ELAVL1, HNRNPD
RBP	Clust B04	AGO1, FMR1_iso7, HNRNPH, LIN28A, LIN28B, TIAL1, ZC3H7B
Singleton	Clust B05	PUM1
RBP	Clust B06	AGO1, AGO2, C22ORF28, CAPRIN1, ELAVL1, EWSR1, FMR1_iso1, FMR1_iso7, FUS, FXR2, HNRNPF, HNRNPU, LIN28A, LIN28B, MOV10, TIA1, TIAL1, ZC3H7B
miRNA	Clust B07	hsa-miR-130a, hsa-miR-130b, hsa-miR-148a, hsa-miR-148b, hsa-miR-19a, hsa-miR-19b, hsa-miR-301a, hsa-miR-301b
Singleton	Clust B08	HNRNPA2B1
Singleton	Clust B09	PABPC1
Singleton	Clust B10	U2AF2
miRNA	Clust B11	hsa-miR-103, hsa-miR-107, hsa-miR-15a, hsa-miR-15b, hsa-miR-16, hsa-miR-22, hsa-miR-29a, hsa-miR-29b, hsa-miR-29c, hsa-miR-424
Singleton	Clust B12	DGCR8
miRNA	Clust B13	hsa-let-7a, hsa-let-7b, hsa-let-7c, hsa-let-7d, hsa-let-7e, hsa-let-7f, hsa-let-7g, hsa-let-7i, hsa-miR-151-5p, hsa-miR-196a, hsa-miR-196b
RBP	Clust B14	AGO1, AGO2, C22ORF28, CAPRIN1, ELAVL1, EWSR1, FMR1_iso1, FMR1_iso7, FUS, FXR1, FXR2, HNRNPF, LIN28A, LIN28B, MOV10, TAF15, TIA1, TIAL1, ZC3H7B
Singleton	Clust B15	C17ORF85
Singleton	Clust B16	TARDBP
Singleton	Clust B17	ALKBH5
miRNA	Clust B18	hsa-miR-106b, hsa-miR-130a, hsa-miR-130b, hsa-miR-148a, hsa-miR-148b, hsa-miR-17, hsa-miR-18a, hsa-miR-20a, hsa-miR-301a, hsa-miR-301b, hsa-miR-320, hsa-miR-93
miRNA	Clust B19	hsa-miR-103, hsa-miR-107, hsa-miR-130a, hsa-miR-130b, hsa-miR-183, hsa-miR-221, hsa-miR-222, hsa-miR-23b, hsa-miR-25, hsa-miR-27a, hsa-miR-27b, hsa-miR-301a, hsa-miR-301b, hsa-miR-32, hsa-miR-92a, hsa-miR-96
RBP	Clust B20	IGF2BP1, IGF2BP2, IGF2BP3, PUM2, QKI
Singleton	Clust B21	RBFOX2
Singleton	Clust B22	CELF1
Singleton	Clust B23	hsa-miR-124
miRNA	Clust B24	hsa-miR-101, hsa-miR-128, hsa-miR-27a, hsa-miR-27b
Singleton	Clust B25	hsa-miR-1

The balanced association score yielded an optimal τ value of 0.25 that is rather different from the unbalanced case where τ was 0.6, because the alternative association score strongly altered the size of the clusters at fixed τ value. Table 6.3 reports the clusters obtained with the balanced association score. In total, they include 88 trans-acting factors (of which 39 RBPs and 49 miRNAs), that represent a grater share with respect to the clusters displayed in Table 6.1 (56 trans-acting factors, of which 32 RBPs and 24 miRNAs). Even though the average cluster size is the same, the balanced association score produced more singleton clusters and, by consequence, few very large clusters. Intuitively, large clusters regulate smaller sets of genes as

they need to be targeted by all trans-acting factors in the cluster. Therefore, the number of genes associated to the clusters obtained using the unbalanced association score (Figure 6.7a) is much higher than the one of the balanced case (Figure 6.7b). The Jaccard similarity, measuring the trans-acting factor overlap among clusters is lower in the balanced case (Figure 6.7c and 6.7d). Also the Gene Ontology enrichments related to clusters from of balanced case are more specific. In fact, the heat maps of the enriched GO terms diplay less overlap with respect to the unbalanced case (Figure 6.7e and 6.7f).

The balanced approach tends to extract clusters formed by trans-acting factors with a similar number of interactions, excluding, for instance, clusters containing both miRNAs and RBPs that emerged from the unbalanced approach. The two procedures allow the discovery of different types of interesting combinatorial patterns present in the data.

6.4 Comparison with related work

PTRcombiner discovers post-transcriptional regulation patterns from interaction maps at a genome-wide level. Other previous attempts have been made to develop automated approaches for the identification of the combinatorial aspects of post-transcriptional gene regulation. In this section, I compare PTRcombiner with PicTar (Krek *et al.*, 2005), ComiR (Coronnello and Benos, 2013) and LeMoNe (Joshi *et al.*, 2008, 2009) highlighting the main differences and providing a further validation of the results obtained by PTRcombiner.

6.4.1 PicTar and ComiR

PicTar computes the probability of multiple miRNAs binding at interim to the same target mRNA. Albeit focusing on combinatorial interactions, PicTar differs from PTRcombiner in many aspects. First, its domain of exploration is limited to miRNAs only. Second, it relies on predicted interactions instead of exploiting experimental data. The last and main caveat of PicTar is that it does not allow to efficiently explore the combinatorial space of possible clusters. In fact, it requires to specify a set of miRNA to be jointly evaluated, that implies the necessity to try all the possible combinations in order to identify high rated clusters. PTRcombiner acts way differently, by implementing an efficient mining procedure, guided by exper-

imental data, that explores the combinatorial space of candidate clusters of miRNAs and/or RBPs.

I analyzed miRNA clusters S09 and S14 (Table 6.2) using PicTar. I focused only on these two clusters because they are composed of four miRNAs, and evaluating bigger clusters with PicTar was computationally too expensive. For each cluster, I considered the set of its target genes, which were the genes interacting with all miRNAs in the cluster, and computed the PicTar interaction score with the cluster for each of the target genes. The score was estimated by considering the maximum value of the product of the binding scores of the single miRNAs (binding scores are taken from Anders *et al.* (2012)). Then, these cluster-target scores were compared with those obtained by running the same procedure on the entire set of 12,713 genes found in Dorina (Anders *et al.*, 2012). For both clusters, the difference between the scores computed on cluster-targets and on the full gene set was statistically significant (Welch's two samples t-test), with a confidence of approximately 0.99. This result confirmed the relevance of the clusters extracted from the experimental data by PTRcombiner.

ComiR is a web tool for combinatorial miRNA target prediction. It aggregates, the scores of the single miRNAs, computed with different scoring approaches. The scores are combined using an SVM that outputs the likelihood that the set of miRNAs binds a specific gene. Similarly to PicTar, the main shortcoming of ComiR is the lack of a mining procedure that proposes putative clusters of miRNAs.

Using ComiR, I analyzed all the miRNA clusters extracted by PTRcombiner, i.e, S09, S14, S16, and S22 (Table 6.2). Also for ComiR, I compared cluster-target scores with scores for the entire set of genes, that this time were identified by all the genes in the ComiR output. For all clusters, the statistical significance of the difference between cluster-target and general scores was confirmed (by Welch's two sample test), with a confidence of approximately 1.0.

6.4.2 LeMoNe

LeMoNe is a probabilistic method for inferring regulatory module networks from expression profiles. This approach is used in Joshi *et al.* (2011) for inferring regulatory networks from both transcriptome and translatome expression profiles in yeast. LeMoNe is able to detect putative regulatory

Table 6.4: Comparison between PTRcombiner clusters and LeMoNe clusters.

PTRcombiner	Components	jaccard	LeMoNe	Components
Clust Y01	Npl3, Pab1, Pub1	1.00	Clust L66	Npl3, Pab1, Pub1
Clust Y02	Scp160, Bfr1	1.00	Clust L24	Scp160, Bfr1
Clust Y03	Npl3, Nrd1, Pab1, Pub1	0.75	Clust L66	Npl3, Pab1, Pub1
Clust Y04	Npl3, Nsr1, Pab1	0.67	Clust L85	Pab1, Nsr1
Clust Y05	Pub1, Scp160, Ypl184c	1.00	Clust L176	Scp160, Pub1, Ypl184c
Clust Y06	Nab2, Npl3	1.00	Clust L38	Npl3, Nab2
Clust Y07	Khd1, Pub1	0.33	Clust L70	Hek2, Pub1
Clust Y08	Nab3, Npl3, Nrd1, Pab1, Pub1	0.83	Clust L02	Npl3, Pab1, Nsr1, Pub1, Nrd1, Nab3
Clust Y10	Bfr1, Pub1, Scp160	1.00	Clust L90	Scp160, Pub1, Bfr1
Clust Y11	Cbc2, Msl5, Npl3, Pab1, Pub1	0.60	Clust L66	Npl3, Pab1, Pub1
Clust Y12	Pub1, Scp160, Sik1	0.50	Clust L90	Scp160, Pub1, Bfr1
Clust Y13	Pub1, Tdh3	0.33	Clust L70	Hek2, Pub1
Clust Y14	Pab1, Puf4	0.50	Clust L05	Gbp2, Npl3, Pab1, Puf4
Clust Y15	Pub1, Puf2	0.33	Clust L08	Ssd1, Scp160, She2, Pub1, Ypl184c, Puf2
Clust Y16	Pab1, Puf3	0.33	Clust L85	Pab1, Nsr1
Clust Y17	Pub1, Puf5	0.33	Clust L70	Hek2, Pub1
Clust Y18	Cbc2, Npl3, Nrd1, Pab1, Pub1	0.60	Clust L66	Npl3, Pab1, Pub1
Clust Y19	Pub1, Vts1	0.33	Clust L32	Nrd1, Vts1
Clust Y20	Cbf5, Npl3, Nrd1, Pab1, Pub1	0.60	Clust L66	Npl3, Pab1, Pub1
Clust Y22	Aco1, Nab2, Pub1, Tdh3	0.75	Clust L09	Nab2, Tdh3, Aco1
Clust Y23	Nab6, Npl3, Pab1, Pub1, Ypl184c	0.63	Clust L03	Npl3, Pab1, Puf3, Nab6, Hrb1, Pub1, Cbc2, Ypl184c
Clust Y24	Pub1, Puf1, Scp160	0.50	Clust L90	Scp160, Pub1, Bfr1
Clust Y25	Nce102, Nrd1, Pub1	0.50	Clust L55	Pub1, Nrd1, Ypl184c

modules that characterize specific biological conditions (i.e. stress conditions), while PTRcombiner aims to achieve a more general purpose: the individuation of combinatorial patterns from genome-wide interactions. It was still intriguing to analyze the relationship between clusters detected by the two methods. To compare LeMoNe with PTRcombiner, I ran PTRcombiner on the yeast dataset employed in Joshi *et al.* (2011).

The dataset contains RIP-chip experiments involving 43 RBPs and 5,118 genes, and it annotates 15,391 interactions, with the interaction matrix sparsity value of 0.07. PTRcombiner clusters, obtained with $\tau = 0.4$, were in agreement with the ones found by LeMoNe. Moreover half of the top 10 clusters found by PTRcombiner were identical to clusters found by LeMoNe (Table 6.4).

Chapter 7

Conclusions

Proteins are key players in several processes occurring in living cells. They are synthesized through the processes of transcription and translation (Crick *et al.*, 1970), where numerous regulatory steps occur to control the amount of proteins expressed in a cell. The main focus of this work was on the study of eukaryotic (mainly human) post-transcriptional regulation. RNA binding proteins (RBPs) and micro RNAs (miRNAs) bind mRNA molecules and modulate several regulatory processes. These are the most studied actors of post-transcriptional regulation. Since the understanding of RNA-protein interactions is an essential point for studying post-transcriptional regulation, many experimental techniques have been developed for detecting such interactions (Marchese *et al.*, 2016). This enabled the generation of an unprecendented source of information for the study of the post-transcriptional gene regulation. Despite the continuous advances in the experimental procedures, these techniques are still far from fully uncovering, on their own, the RNA-protein interaction mechanism. I underlined three shortcomings of the data produced by these experimental techniques: first, the available interaction data covers a small fraction of the known RBPs; second, experimentally determined interactions are often noisy and cell-line dependent; and third, these techniques do not provide information of combinatorial interaction of RBPs with the same set of mRNAs.

Computational techniques capable of learning from the data, such as machine learning approaches, are able to generalize the information contained in the data and might give useful insights to help the investigation of

post-transcriptional regulation. In this trandisciplinary thesis, I proposed three machine learning contributions, that address these three mentioned shortcomings of the data obtained with available experimental techniques.

In Chapter 4 I presented RNAcommender, a tool for recommending RNA-protein interactions. By representing RBPs and RNAs with explicit features that account for protein domain composition and RNA secondary structure, and by exploiting the available experimental evidence, RNAcommender enabled the recommendation of RNA targets to RBPs that lack high-throughput experimental evidence of interaction. RNAcommender was validated on a dataset of human RNA-protein interactions, exhibiting good performance in ranking candidate RNA interactors for an RBP (average AUC ROC of 0.75), and a significant enrichment in valid targets in the top 50 predictions for 75% of the tested proteins. RNAcommender can be a valid assistant to experimental research, especially for the investigation of the RBPs whose RNA targets have not yet been experimentally identified or that cannot be identified with such techniques (e.g. RBPs that do not crosslink). For sure, the high complexity of RNA regulation necessitates additional efforts to improve the quality of the predictions. Although protein-protein interactions are known to affect the recognition of RNA substrates (Glisovic *et al.*, 2008), presently RNAcommender does not account for this type of interactions. In the future it would be interesting to modify the model of RNAcommender in order to include protein-protein interactions in the information used to recommend RNA targets.

In Chapter 5 I presented ProtScan, a tool based on consensus kernelized SGD regression for effective modeling of RNA-protein interactions. ProtScan outperformed competitor state-of-the-art methods, proving a powerful tool to model and predict RNA-protein interactions on a transcriptome-wide scale. ProtScan allows to denoise and generalize the information contained in CLIP-seq experiments in order to predict interaction profiles for RBPs at a genome-wide scale. Moreover, ProtScan includes a peak detection technique that automatically extracts predicted binding regions from the generated interaction profiles. ProtScan is an helpful tool that should be taken into account to post-process high-throughput experiments in order to remove the experimental noise present in the obtained data. To further improve the performance of ProtScan, future work might be done for including types of information that are known to be associated with RBP

binding. Some examples are mRNA accessibility and the presence of target sites for regulatory entities such as miRNAs and other known competitive or cooperative RBPs.

In Chapter 6 I presented PTRcombiner, a tool for the discovery and analysis of post-transcriptional regulation patterns involving multiple trans-factors. PTRcombiner was tested on experimental interaction information between post-transcriptional trans-factors and their respective targets, obtained in both human and yeast. This tool enabled the detection of groups of regulators that share a conspicuous amount of targets; the biological characterization of the clusters; and the identification of potential concurrent binding sites for RBPs belonging to the same cluster. PTRcombiner represents an original and comprehensive attempt to implement a computational pipeline for decoding complex post-transcriptional combinatorial rules at a genome-wide scale. A future improvement that might be worth exploring is the relaxation of the Boolean constraints on the input data. This would allow to integrate information regarding expression profiles of both trans-acting factors and target mRNAs allowing to mine combinatorial patterns in specific experimental conditions. Moreover, the relaxation of the Boolean constraints will allow to also deal with uncertainty of the interaction information, such as predited interactions from tools like RNAcommender.

In conclusion, the main aim of this transdisciplinary research work was to release tools that might assist the investigation of the post-transcriptional gene regulation. The hope is that in the near future these research contributions will prove to be valuable assistants to researchers and help to unveil some yet uncharacterized aspects of the post-transcriptional gene regulation.

Bibliography

Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE transactions on knowledge and data engineering*, **17**(6), 734–749.

Aggarwal, C. C. (2016). *Recommender Systems: The Textbook*. Springer.

Agostini, F., Zanzoni, A., Klus, P., Marchese, D., Cirillo, D., and Tartaglia, G. G. (2013). catRAPID omics: a web server for large-scale prediction of protein–RNA interactions. *Bioinformatics*, **29**(22), 2928–2930.

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM.

Aizerman, A., Braverman, E. M., and Rozoner, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and remote control*, **25**, 821–837.

Aken, B. L., Ayling, S., Barrell, D., Clarke, L., Curwen, V., Fairley, S., Banet, J. F., Billis, K., Girón, C. G., Hourlier, T., *et al.* (2016). The Ensembl gene annotation system. *Database*, **2016**, baw093.

Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P. (2002). *Molecular Biology of the Cell, Fourth Edition*. Garland Science, 4 edition.

Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. *Nature biotechnology*, **33**(8), 831–838.

Ament, S. A., Blatti, C. A., Alaux, C., Wheeler, M. M., Toth, A. L., Le Conte, Y., Hunt, G. J., Guzmán-Novoa, E., DeGrandi-Hoffman, G., Uribe-Rubio, J. L., *et al.* (2012). New meta-analysis tools reveal common transcriptional regulatory basis for multiple determinants of behavior. *Proceedings of the National Academy of Sciences*, **109**(26), E1801–E1810.

Anders, G., Mackowiak, S. D., Jens, M., Maaskola, J., Kuntzagk, A., Rajewsky, N., Landthaler, M., and Dieterich, C. (2012). doRiNA: a database of RNA interactions in post-transcriptional regulation. *Nucleic acids research*, **40**(D1), D180–D186.

Andersson, M. K., Ståhlberg, A., Arvidsson, Y., Olofsson, A., Semb, H., Stenman, G., Nilsson, O., and Åman, P. (2008). The multifunctional FUS, EWS and TAF15 proto-oncoproteins show cell type-specific expression patterns and involvement in cell spreading and stress response. *BMC cell biology*, **9**(1), 1.

Arcondéguy, T., Lacazette, E., Millevoi, S., Prats, H., and Touriol, C. (2013). VEGF-A mRNA processing, stability and translation: a paradigm for intricate regulation of gene expression at the post-transcriptional level. *Nucleic acids research*, **41**(17), 7997–8010.

Ascano, M., Mukherjee, N., Bandaru, P., Miller, J. B., Nusbaum, J. D., Corcoran, D. L., Langlois, C., Munschauer, M., Dewell, S., Hafner, M., *et al.* (2012). FMRP targets distinct mRNA sequence elements to regulate protein expression. *Nature*, **492**(7429), 382–386.

Asif, H. S. and Sanguinetti, G. (2011). Large-scale learning of combinatorial transcriptional dynamics from gene expression. *Bioinformatics*, **27**(9), 1277–1283.

Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., Ren, J., Li, W. W., and Noble, W. S. (2009). MEME SUITE: tools for motif discovery and searching. *Nucleic acids research*, page gkp335.

Bailly-Bechet, M., Braunstein, A., Pagnani, A., Weigt, M., and Zecchina, R. (2010). Inference of sparse combinatorial-control networks from gene-expression data: a message passing approach. *BMC bioinformatics*, **11**(1), 1.

Balabanović, M. and Shoham, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*, **40**(3), 66–72.

Baltz, A. G., Munschauer, M., Schwanhäusser, B., Vasile, A., Murakawa, Y., Schueler, M., Youngs, N., Penfold-Brown, D., Drew, K., Milek, M., *et al.* (2012). The mRNA-bound proteome and its global occupancy profile on protein-coding transcripts. *Molecular cell*, **46**(5), 674–690.

Barrass, J. D., Reid, J. E., Huang, Y., Hector, R. D., Sanguinetti, G., Beggs, J. D., and Granneman, S. (2015). Transcriptome-wide RNA processing kinetics revealed using extremely short 4tU labeling. *Genome biology*, **16**(1), 1.

Beilharz, T. H. and Preiss, T. (2007). Widespread use of poly (A) tail length control to accentuate expression of the yeast transcriptome. *Rna*, **13**(7), 982–997.

Bellucci, M., Agostini, F., Masin, M., and Tartaglia, G. G. (2011). Predicting protein associations with long noncoding RNAs. *Nature methods*, **8**(6), 444–445.

Bergstra, J. and Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, **13**(Feb), 281–305.

Billsus, D. and Pazzani, M. J. (1998). Learning collaborative information filters. In *Icml*, volume 98, pages 46–54.

Billsus, D. and Pazzani, M. J. (2000). User modeling for adaptive news access. *User modeling and user-adapted interaction*, **10**(2-3), 147–180.

Blaxall, B. C., Pende, A., Wu, S. C., and Port, J. D. (2002). Correlation between intrinsic mRNA stability and the affinity of AUF1 (huRNP D) and HuR for A+ U-rich mRNAs. *Molecular and cellular biochemistry*, **232**(1-2), 1–11.

Blin, K., Dieterich, C., Wurmus, R., Rajewsky, N., Landthaler, M., and Akalin, A. (2015). DoRiNa 2.0—upgrading the doRiNa database of RNA interactions in post-transcriptional regulation. *Nucleic acids research*, **43**(D1), D160–D167.

Bolger, A. M., Lohse, M., and Usadel, B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, page btu170.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Bossy-Wetzel, E., Schwarzenbacher, R., and Lipton, S. A. (2004). Molecular pathways to neurodegeneration.

Braun, J. E., Huntzinger, E., and Izaurralde, E. (2012). A molecular link between miRISCs and deadenylases provides new insight into the mechanism of gene silencing by microRNAs. *Cold Spring Harbor perspectives in biology*, **4**(12), a012328.

Bray, N. L., Pimentel, H., Melsted, P., and Pachter, L. (2016). Near-optimal probabilistic RNA-seq quantification. *Nature biotechnology*, **34**(5), 525–527.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pages 43–52. Morgan Kaufmann Publishers Inc.

Breiman, L. (2001). Random forests. *Machine learning*, **45**(1), 5–32.

Burd, C. G. and Dreyfuss, G. (1994). RNA binding specificity of hnRNP A1: significance of hnRNP A1 high-affinity binding sites in pre-mRNA splicing. *The EMBO journal*, **13**(5), 1197.

Calvo, S. E., Pagliarini, D. J., and Mootha, V. K. (2009). Upstream open reading frames cause widespread reduction of protein expression and are polymorphic among humans. *Proceedings of the National Academy of Sciences*, **106**(18), 7507–7512.

Castello, A., Fischer, B., Eichelbaum, K., Horos, R., Beckmann, B. M., Strein, C., Davey, N. E., Humphreys, D. T., Preiss, T., Steinmetz, L. M., *et al.* (2012). Insights into RNA biology from an atlas of mammalian mRNA-binding proteins. *Cell*, **149**(6), 1393–1406.

Chen, E., Sharma, M. R., Shi, X., Agrawal, R. K., and Joseph, S. (2014). Fragile X mental retardation protein regulates translation by binding directly to the ribosome. *Molecular cell*, **54**(3), 407–417.

Chesler, E. J. and Langston, M. A. (2007). Combinatorial genetic regulatory network analysis tools for high throughput transcriptomic data. In *Systems Biology and Regulatory Genomics*, pages 150–165. Springer.

Chi, S. W., Zang, J. B., Mele, A., and Darnell, R. B. (2009). Argonaute HITS-CLIP decodes microRNA–mRNA interaction maps. *Nature*, **460**(7254), 479–486.

Christakou, C., Vrettos, S., and Stafylopatis, A. (2007). A hybrid movie recommender system based on neural networks. *International Journal on Artificial Intelligence Tools*, **16**(05), 771–792.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on pattern analysis and machine intelligence*, **24**(5), 603–619.

Consortium, E. P. *et al.* (2004). The encode (encyclopedia of dna elements) project. *Science*, **306**(5696), 636–640.

Corcoran, D. L., Georgiev, S., Mukherjee, N., Gottwein, E., Skalsky, R. L., Keene, J. D., and Ohler, U. (2011). PARalyzer: definition of RNA binding sites from PAR-CLIP short-read sequence data. *Genome biology*, **12**(8), 1.

Coronnello, C. and Benos, P. V. (2013). ComiR: combinatorial microRNA target prediction tool. *Nucleic acids research*, **41**(W1), W159–W164.

Costa, F. and De Grave, K. (2010). Fast neighborhood subgraph pairwise distance kernel. In *Proceedings of the 26th International Conference on Machine Learning*, pages 255–262. Omnipress.

Craig, A. W., Haghighat, A., Annie, T., and Sonenberg, N. (1998). Interaction of polyadenylate-binding protein with the eIF4G homologue PAIP enhances translation. *Nature*, **392**(6675), 520–523.

Crick, F. *et al.* (1970). Central dogma of molecular biology. *Nature*, **227**(5258), 561–563.

Culjkovic-Kraljacic, B. and Borden, K. L. (2013). Aiding and abetting cancer: mRNA export and the nuclear pore. *Trends in cell biology*, **23**(7), 328–335.

Curk, T., König, J., Gorup, Č., Rot, G., Ule, J., and Zupan, B. (2011). Comprehensive analysis of iCLIP high-throughput sequencing data with iCount. *6th CFG nd Fig*, page 51.

Dassi, E., Re, A., Leo, S., Tebaldi, T., Pasini, L., Peroni, D., and Quattrone, A. (2014). AURA 2: empowering discovery of post-transcriptional networks. *Translation*, **2**(1), e27738.

Des Georges, A., Dhote, V., Kuhn, L., Hellen, C. U., Pestova, T. V., Frank, J., and Hashem, Y. (2015). Structure of mammalian eIF3 in the context of the 43S preinitiation complex. *Nature*.

Deshpande, M. and Karypis, G. (2004). Item-based top-n recommendation algorithms. *ACM Transactions on Information Systems (TOIS)*, **22**(1), 143–177.

Di Giammartino, D. C., Nishida, K., and Manley, J. L. (2011). Mechanisms and consequences of alternative polyadenylation. *Molecular cell*, **43**(6), 853–866.

Ding, C., Li, T., Peng, W., and Park, H. (2006). Orthogonal nonnegative matrix t-factorizations for clustering. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 126–135. ACM.

Dominguez, C., Schubert, M., Duss, O., Ravindranathan, S., and Allain, F. H.-T. (2011). Structure determination and dynamics of protein–RNA complexes by NMR spectroscopy. *Progress in nuclear magnetic resonance spectroscopy*, **58**(1), 1–61.

Eichhorn, S. W., Guo, H., McGeary, S. E., Rodriguez-Mias, R. A., Shin, C., Baek, D., Hsu, S.-h., Ghoshal, K., Villén, J., and Bartel, D. P. (2014). mRNA destabilization is the dominant effect of mammalian microRNAs by the time substantial repression ensues. *Molecular cell*, **56**(1), 104–115.

El Baroudi, M., Corà, D., Bosia, C., Osella, M., and Caselle, M. (2011). A curated database of miRNA mediated feed-forward loops involving MYC as master regulator. *PloS one*, **6**(3), e14742.

Ellington, A. D. and Szostak, J. W. (1990). In vitro selection of RNA molecules that bind specific ligands. *Nature*, **346**(6287), 818–822.

Fabian, M. R. and Sonenberg, N. (2012). The mechanics of miRNA-mediated gene silencing: a look under the hood of miRISC. *Nature structural & molecular biology*, **19**(6), 586–593.

Farazi, T. A., Spitzer, J. I., Morozov, P., and Tuschl, T. (2011). miRNAs in human cancer. *The Journal of pathology*, **223**(2), 102–115.

Ferrarese, R., Harsh, G. R., Yadav, A. K., Bug, E., Maticzka, D., Reichardt, W., Dombrowski, S. M., Miller, T. E., Masilamani, A. P., Dai, F., *et al.* (2014). Lineage-specific splicing of a brain-enriched alternative exon promotes glioblastoma progression. *The Journal of clinical investigation*, **124**(7), 2861–2876.

Finn, R. D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Heger, A., Hetherington, K., Holm, L., Mistry, J., *et al.* (2013). Pfam: the protein families database. *Nucleic acids research*, page gkt1223.

Foat, B. C., Morozov, A. V., and Bussemaker, H. J. (2006). Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. *Bioinformatics*, **22**(14), e141–e149.

Friard, O., Re, A., Taverna, D., De Bortoli, M., and Corá, D. (2010). CircuitsDB: a database of mixed microRNA/transcription factor feed-forward regulatory circuits in human and mouse. *BMC bioinformatics*, **11**(1), 435.

Galgano, A., Forrer, M., Jaskiewicz, L., Kanitz, A., Zavolan, M., and Gerber, A. P. (2008). Comparative analysis of mRNA targets for human PUF-family proteins suggests extensive interaction with the miRNA regulatory system. *PloS one*, **3**(9), e3164.

Geerts, F., Goethals, B., and Mielikäinen, T. (2004). Tiling databases. In *International Conference on Discovery Science*, pages 278–289. Springer.

Gerstberger, S., Hafner, M., and Tuschl, T. (2014). A census of human RNA-binding proteins. *Nature Reviews Genetics*, **15**(12), 829–845.

Geuens, T., Bouhy, D., and Timmerman, V. (2016). The hnRNP family: insights into their role in health and disease. *Human genetics*, pages 1–17.

Giegerich, R., Voß, B., and Rehmsmeier, M. (2004). Abstract shapes of RNA. *Nucleic acids research*, **32**(16), 4843–4851.

Giménez-Barcons, M. and Díez, J. (2011). Yeast processing bodies and stress granules: self-assembly ribonucleoprotein particles. *Microbial cell factories*, **10**(1), 1.

Glisovic, T., Bachorik, J. L., Yong, J., and Dreyfuss, G. (2008). RNA-binding proteins and post-transcriptional gene regulation. *FEBS letters*, **582**(14), 1977–1986.

Granneman, S., Kudla, G., Petfalski, E., and Tollervey, D. (2009). Identification of protein binding sites on U3 snoRNA and pre-rRNA by UV cross-linking and high-throughput analysis of cDNAs. *Proceedings of the National Academy of Sciences*, **106**(24), 9613–9618.

Gupta, S. K., Kosti, I., Plaut, G., Pivko, A., Tkacz, I. D., Cohen-Chalamish, S., Biswas, D. K., Wachtel, C., Ben-Asher, H. W., Carmi, S., *et al.* (2013). The hnRNP F/H homologue of Trypanosoma brucei is differentially expressed in the two life cycle stages of the parasite and regulates splicing and mRNA stability. *Nucleic acids research*, **41**(13), 6577–6594.

Hafner, M., Landthaler, M., Burger, L., Khorshid, M., Hausser, J., Berninger, P., Rothballer, A., Ascano, M., Jungkamp, A.-C., Munschauer, M., *et al.* (2010). Transcriptome-wide identification of RNA-binding protein and microRNA target sites by PAR-CLIP. *Cell*, **141**(1), 129–141.

Hansen, L. K. and Salamon, P. (1990). Neural network ensembles. *IEEE transactions on pattern analysis and machine intelligence*, **12**, 993–1001.

Haussler, D. (1999). Convolution kernels on discrete structures. Technical report, Citeseer.

Hellman, L. M. and Fried, M. G. (2007). Electrophoretic mobility shift assay (EMSA) for detecting protein–nucleic acid interactions. *Nature protocols*, **2**(8), 1849–1861.

Hennig, J. and Sattler, M. (2015). Deciphering the protein-RNA recognition code: Combining large-scale quantitative methods with structural biology. *BioEssays*, **37**(8), 899–908.

Heyne, S., Costa, F., Rose, D., and Backofen, R. (2012). GraphClust: alignment-free structural clustering of local RNA secondary structures. *Bioinformatics*, **28**(12), i224–i232.

Hiller, M., Pudimat, R., Busch, A., and Backofen, R. (2006). Using RNA secondary structures to guide sequence motif finding towards single-stranded regions. *Nucleic acids research*, **34**(17), e117–e117.

Hoerl, A. E. and Kennard, R. W. (1970). Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, **12**(1), 55–67.

Howe, A. E. and Forbes, R. D. (2008). Re-considering neighborhood-based collaborative filtering parameters in the context of new data. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 1481–1482. ACM.

Hu, B., Yang, Y.-C. T., Huang, Y., Zhu, Y., and Lu, Z. J. (2016). POSTAR: a platform for exploring post-transcriptional regulation coordinated by RNA-binding proteins. *Nucleic Acids Research*, page gkw888.

Jaakkola, T., Diekhans, M., and Haussler, D. (2000). A discriminative framework for detecting remote protein homologies. *Journal of computational biology*, **7**(1-2), 95–114.

Jankowsky, E. and Harris, M. E. (2015). Specificity and nonspecificity in RNA-protein interactions. *Nature Reviews Molecular Cell Biology*, **16**(9), 533–544.

Jiang, P., Singh, M., and Coller, H. A. (2013). Computational assessment of the cooperativity between RNA binding proteins and MicroRNAs in Transcript Decay. *PLoS Comput Biol*, **9**(5), e1003075.

Joshi, A., Van de Peer, Y., and Michoel, T. (2008). Analysis of a Gibbs sampler method for model-based clustering of gene expression data. *Bioinformatics*, **24**(2), 176–183.

Joshi, A., De Smet, R., Marchal, K., Van de Peer, Y., and Michoel, T. (2009). Module networks revisited: computational assessment and prioritization of model predictions. *Bioinformatics*, **25**(4), 490–496.

Joshi, A., Van de Peer, Y., and Michoel, T. (2011). Structural and functional organization of RNA regulons in the post-transcriptional regulatory network of yeast. *Nucleic acids research*, **39**(21), 9108–9117.

Karlebach, G. and Shamir, R. (2008). Modelling and analysis of gene regulatory networks. *Nature Reviews Molecular Cell Biology*, **9**(10), 770–780.

Kazan, H., Ray, D., Chan, E. T., Hughes, T. R., and Morris, Q. (2010). RNAcontext: a new method for learning the sequence and structure binding preferences of RNA-binding proteins. *PLoS Comput Biol*, **6**(7), e1000832.

Ke, A. and Doudna, J. A. (2004). Crystallization of RNA and RNA–protein complexes. *Methods*, **34**(3), 408–414.

Keene, J. D. (2007). RNA regulons: coordination of post-transcriptional events. *Nature Reviews Genetics*, **8**(7), 533–543.

Kemény-Beke, A., Berényi, E., Facskó, A., Damjanovich, J., Horváth, A., Bodnár, A., Berta, A., and Aradi, J. (2006). Antiproliferative effect of 4-thiouridylate on OCM-1 uveal melanoma cells. *European journal of ophthalmology*, **16**(5), 680.

Kim, H. H., Kuwano, Y., Srikantan, S., Lee, E. K., Martindale, J. L., and Gorospe, M. (2009). HuR recruits let-7/RISC to repress c-Myc expression. *Genes & development*, **23**(15), 1743–1748.

König, J., Zarnack, K., Rot, G., Curk, T., Kayikci, M., Zupan, B., Turner, D. J., Luscombe, N. M., and Ule, J. (2010). iCLIP reveals the function of hnRNP particles in splicing at individual nucleotide resolution. *Nature structural & molecular biology*, **17**(7), 909–915.

Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, **40**(3), 77–87.

Koren, Y., Bell, R., Volinsky, C., *et al.* (2009). Matrix factorization techniques for recommender systems. *Computer*, **42**(8), 30–37.

Krek, A., Grün, D., Poy, M. N., Wolf, R., Rosenberg, L., Epstein, E. J., MacMenamin, P., Da Piedade, I., Gunsalus, K. C., Stoffel, M., *et al.* (2005). Combinatorial microRNA target predictions. *Nature genetics*, **37**(5), 495–500.

Kudla, G., Granneman, S., Hahn, D., Beggs, J. D., and Tollervey, D. (2011). Cross-linking, ligation, and sequencing of hybrids reveals RNA–RNA interactions in yeast. *Proceedings of the National Academy of Sciences*, **108**(24), 10010–10015.

Lambert, N., Robertson, A., Jangi, M., McGeary, S., Sharp, P. A., and Burge, C. B. (2014). RNA Bind-n-Seq: quantitative assessment of the sequence and structural binding specificity of RNA binding proteins. *Molecular cell*, **54**(5), 887–900.

Landthaler, M., Gaidatzis, D., Rothballer, A., Chen, P. Y., Soll, S. J., Dinic, L., Ojo, T., Hafner, M., Zavolan, M., and Tuschl, T. (2008). Molecular characterization of human Argonaute-containing ribonucleoprotein complexes and their bound target mRNAs. *Rna*, **14**(12), 2580–2596.

Lange, S. J., Maticzka, D., Möhl, M., Gagnon, J. N., Brown, C. M., and Backofen, R. (2012). Global or local? Predicting secondary structure and accessibility in mRNAs. *Nucleic acids research*, page gks181.

Langmead, B. and Salzberg, S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nature methods*, **9**(4), 357–359.

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome biology*, **10**(3), 1.

Lebedeva, S., Jens, M., Theil, K., Schwanhäusser, B., Selbach, M., Landthaler, M., and Rajewsky, N. (2011). Transcriptome-wide analysis of regulatory interactions of the RNA-binding protein HuR. *Molecular cell*, **43**(3), 340–352.

Leslie, C. S., Eskin, E., and Noble, W. S. (2002). The spectrum kernel: A string kernel for SVM protein classification. In *Pacific symposium on biocomputing*, volume 7, pages 566–575.

Leslie, C. S., Eskin, E., Cohen, A., Weston, J., and Noble, W. S. (2004). Mismatch string kernels for discriminative protein classification. *Bioinformatics*, **20**(4), 467–476.

Li, H., Xuan, J., Wang, Y., and Zhan, M. (2008). Inferring regulatory networks. *Front Biosci*, **13**(263), 75.

Li, P. and König, C. (2010). b-Bit minwise hashing. In *Proceedings of the 19th international conference on World wide web*, pages 671–680. ACM.

Licatalosi, D. D., Mele, A., Fak, J. J., Ule, J., Kayikci, M., Chi, S. W., Clark, T. A., Schweitzer, A. C., Blume, J. E., Wang, X., *et al.* (2008). HITS-CLIP yields genome-wide insights into brain alternative RNA processing. *Nature*, **456**(7221), 464–469.

Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*, **7**(1), 76–80.

Lorenz, R., Bernhart, S. H., Zu Siederdissen, C. H., Tafer, H., Flamm, C., Stadler, P. F., and Hofacker, I. L. (2011). ViennaRNA Package 2.0. *Algorithms for Molecular Biology*, **6**(1), 1.

Lunde, B. M., Moore, C., and Varani, G. (2007). RNA-binding proteins: modular design for efficient function. *Nature reviews Molecular cell biology*, **8**(6), 479–490.

Manche, L., Green, S. R., Schmedt, C., and Mathews, M. B. (1992). Interactions between double-stranded RNA regulators and the protein kinase DAI. *Molecular and cellular biology*, **12**(11), 5238–5248.

Marchese, D., de Groot, N. S., Lorenzo Gotor, N., Livi, C. M., and Tartaglia, G. G. (2016). Advances in the characterization of RNA-binding proteins. *Wiley Interdisciplinary Reviews: RNA*, **7**(6), 793–810.

Maticzka, D., Lange, S. J., Costa, F., and Backofen, R. (2014). GraphProt: modeling binding preferences of RNA-binding proteins. *Genome biology*, **15**(1), 1.

Miettinen, P., Mielikäinen, T., Gionis, A., Das, G., and Mannila, H. (2008). The discrete basis problem. *IEEE Transactions on Knowledge and Data Engineering*, **20**(10), 1348–1362.

Mili, S. and Steitz, J. A. (2004). Evidence for reassociation of RNA-binding proteins after cell lysis: implications for the interpretation of immunoprecipitation analyses. *Rna*, **10**(11), 1692–1694.

Miyahara, K. and Pazzani, M. J. (2000). Collaborative filtering with the simple Bayesian classifier. In *Pacific Rim International conference on artificial intelligence*, pages 679–689. Springer.

Modic, M., Ule, J., and Sibley, C. R. (2013). CLIPing the brain: studies of protein–RNA interactions important for neurodegenerative disorders. *Molecular and Cellular Neuroscience*, **56**, 429–435.

Mukherjee, N., Corcoran, D. L., Nusbaum, J. D., Reid, D. W., Georgiev, S., Hafner, M., Ascano, M., Tuschl, T., Ohler, U., and Keene, J. D. (2011). Integrative regulatory mapping indicates that the RNA-binding protein HuR couples pre-mRNA processing and mRNA stability. *Molecular cell*, **43**(3), 327–339.

Muppirala, U. K., Honavar, V. G., and Dobbs, D. (2011). Predicting RNA-protein interactions using only sequence information. *BMC bioinformatics*, **12**(1), 1.

Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 809–816.

Pancaldi, V. and Bähler, J. (2011). In silico characterization and prediction of global protein–mRNA interactions in yeast. *Nucleic acids research*, **39**(14), 5826–5836.

Pasquinelli, A. E. (2012). MicroRNAs and their targets: recognition, regulation and an emerging reciprocal relationship. *Nature Reviews Genetics*, **13**(4), 271–282.

Patel, V. L., Mitra, S., Harris, R., Buxbaum, A. R., Lionnet, T., Brenowitz, M., Girvin, M., Levy, M., Almo, S. C., Singer, R. H., *et al.* (2012). Spatial arrangement of an RNA zipcode identifies mRNAs under post-transcriptional control. *Genes & Development*, **26**(1), 43–53.

Polson, A. G. and Bass, B. L. (1994). Preferential selection of adenosines for modification by double-stranded RNA adenosine deaminase. *The EMBO journal*, **13**(23), 5701.

Ray, D., Kazan, H., Chan, E. T., Castillo, L. P., Chaudhry, S., Talukder, S., Blencowe, B. J., Morris, Q., and Hughes, T. R. (2009). Rapid and systematic analysis of the RNA recognition specificities of RNA-binding proteins. *Nature biotechnology*, **27**(7), 667–670.

Ray, D., Kazan, H., Cook, K. B., Weirauch, M. T., Najafabadi, H. S., Li, X., Gueroussov, S., Albu, M., Zheng, H., Yang, A., *et al.* (2013). A compendium of RNA-binding motifs for decoding gene regulation. *Nature*, **499**(7457), 172–177.

Re, A., Corá, D., Taverna, D., and Caselle, M. (2009). Genome-wide survey of microRNA–transcription factor feed-forward regulatory circuits in human. *Molecular BioSystems*, **5**(8), 854–867.

Rehwinkel, J., Behm-Ansmant, I., Gatfield, D., and Izaurralde, E. (2005). A crucial role for GW182 and the DCP1: DCP2 decapping complex in miRNA-mediated gene silencing. *Rna*, **11**(11), 1640–1647.

Richter, J. D., Bassell, G. J., and Klann, E. (2015). Dysregulation and restoration of translational homeostasis in fragile X syndrome. *Nature Reviews Neuroscience*.

Rodriguez, J. M., Carro, A., Valencia, A., and Tress, M. L. (2015). APPRIS WebServer and WebServices. *Nucleic acids research*, **43**(W1), W455–W459.

Rose, P. W., Prlić, A., Bi, C., Bluhm, W. F., Christie, C. H., Dutta, S., Green, R. K., Goodsell, D. S., Westbrook, J. D., Woo, J., *et al.* (2015). The RCSB Protein Data Bank: views of structural biology for basic and applied research and education. *Nucleic acids research*, **43**(D1), D345–D356.

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*, pages 791–798. ACM.

Sanford, J. R., Wang, X., Mort, M., VanDuyn, N., Cooper, D. N., Mooney, S. D., Edenberg, H. J., and Liu, Y. (2009). Splicing factor SFRS1 recognizes a functionally diverse landscape of RNA transcripts. *Genome research*, **19**(3), 381–394.

Schoenberg, D. R. and Maquat, L. E. (2012). Regulation of cytoplasmic mRNA decay. *Nature Reviews Genetics*, **13**(4), 246–259.

Shani, G. and Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender systems handbook*, pages 257–297. Springer.

Shapira, B., Ricci, F., Kantor, P. B., and Rokach, L. (2011). Recommender Systems Handbook.

Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating "word of mouth". In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 210–217. ACM Press/Addison-Wesley Publishing Co.

Shawe-Taylor, J. and Cristianini, N. (2004). *Kernel methods for pattern analysis*. Cambridge university press.

Simone, L. E. and Keene, J. D. (2013). Mechanisms coordinating ELAV/Hu mRNA regulons. *Current opinion in genetics & development*, **23**(1), 35–43.

Sloan, C. A., Chan, E. T., Davidson, J. M., Malladi, V. S., Strattan, J. S., Hitz, B. C., Gabdank, I., Narayanan, A. K., Ho, M., Lee, B. T., *et al.* (2016). ENCODE data at the ENCODE portal. *Nucleic acids research*, **44**(D1), D726–D732.

Speir, M. L., Zweig, A. S., Rosenbloom, K. R., Raney, B. J., Paten, B., Nejad, P., Lee, B. T., Learned, K., Karolchik, D., Hinrichs, A. S., *et al.* (2016). The UCSC Genome Browser database: 2016 update. *Nucleic acids research*, **44**(D1), D717–D725.

St Johnston, D., Brown, N. H., Gall, J. G., and Jantsch, M. (1992). A conserved double-stranded RNA-binding domain. *Proceedings of the National Academy of Sciences*, **89**(22), 10979–10983.

Sugimoto, Y., Vigilante, A., Darbo, E., Zirra, A., Militti, C., D'Ambrogio, A., Luscombe, N. M., and Ule, J. (2015). hiCLIP reveals the in vivo atlas of mRNA secondary structures recognized by Staufen 1. *Nature*, **519**(7544), 491–494.

Takács, G., Pilászy, I., Németh, B., and Tikk, D. (2008). Investigation of various matrix factorization methods for large recommender systems. In *2008 IEEE International Conference on Data Mining Workshops*, pages 553–562. IEEE.

Takács, G., Pilászy, I., Németh, B., and Tikk, D. (2009). Scalable collaborative filtering approaches for large recommender systems. *Journal of machine learning research*, **10**(Mar), 623–656.

Tatti, N. and Heikinheimo, H. (2008). Decomposable families of itemsets. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 472–487. Springer.

Tatti, N. and Vreeken, J. (2008). Finding good itemsets by packing data. In *2008 Eighth IEEE International Conference on Data Mining*, pages 588–597. IEEE.

Tebaldi, T., Re, A., Viero, G., Pegoretti, I., Passerini, A., Blanzieri, E., and Quattrone, A. (2012). Widespread uncoupling between transcriptome and translatome variations after a stimulus in mammalian cells. *BMC genomics*, **13**(1), 1.

Tenenbaum, S. A., Carson, C. C., Lager, P. J., and Keene, J. D. (2000). Identifying mRNA subsets in messenger ribonucleoprotein complexes by using cDNA arrays. *Proceedings of the National Academy of Sciences*, **97**(26), 14085–14090.

Tian, B. and Manley, J. L. (2017). Alternative polyadenylation of mRNA precursors. *Nature Reviews Molecular Cell Biology*, **18**(1), 18–30.

Tiemann, M. and Pauws, S. (2007). Towards ensemble learning for hybrid music recommendation. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 177–178. ACM.

Tome, J. M., Ozer, A., Pagano, J. M., Gheba, D., Schroth, G. P., and Lis, J. T. (2014). Comprehensive analysis of RNA-protein interactions by high throughput sequencing-RNA affinity profiling. *Nature methods*, **11**(6), 683.

Ule, J., Jensen, K. B., Ruggiu, M., Mele, A., Ule, A., and Darnell, R. B. (2003). CLIP identifies Nova-regulated RNA networks in the brain. *Science*, **302**(5648), 1212–1215.

Uren, P. J., Bahrami-Samani, E., Burns, S. C., Qiao, M., Karginov, F. V., Hodges, E., Hannon, G. J., Sanford, J. R., Penalva, L. O., and Smith, A. D. (2012). Site identification in high-throughput RNA–protein interaction data. *Bioinformatics*, **28**(23), 3013–3020.

Van Nostrand, E. L., Pratt, G. A., Shishkin, A. A., Gelboin-Burkhart, C., Fang, M. Y., Sundararaman, B., Blue, S. M., Nguyen, T. B., Surka, C., Elkins, K., *et al.* (2016). Robust transcriptome-wide discovery of RNA-binding protein binding sites with enhanced CLIP (eCLIP). *Nature methods*, **13**(6), 508–514.

Verkerk, A. J., Pieretti, M., Sutcliffe, J. S., Fu, Y.-H., Kuhl, D. P., Pizzuti, A., Reiner, O., Richards, S., Victoria, M. F., Zhang, F., *et al.* (1991). Identification of a gene (FMR-1) containing a CGG repeat coincident with a breakpoint cluster region exhibiting length variation in fragile X syndrome. *Cell*, **65**(5), 905–914.

Vessey, J. P., Vaccani, A., Xie, Y., Dahm, R., Karra, D., Kiebler, M. A., and Macchi, P. (2006). Dendritic localization of the translational repressor Pumilio 2 and its contribution to dendritic stress granules. *The Journal of neuroscience*, **26**(24), 6496–6508.

Vessey, J. P., Schoderboeck, L., Gingl, E., Luzi, E., Riefler, J., Di Leva, F., Karra, D., Thomas, S., Kiebler, M. A., and Macchi, P. (2010). Mammalian Pumilio 2 regulates dendrite morphogenesis and synaptic function. *Proceedings of the National Academy of Sciences*, **107**(7), 3222–3227.

Vogel, C., de Sousa Abreu, R., Ko, D., Le, S.-Y., Shapiro, B. A., Burns, S. C., Sandhu, D., Boutz, D. R., Marcotte, E. M., and Penalva, L. O. (2010). Sequence signatures and mRNA concentration can explain two-thirds of protein abundance variation in a human cell line. *Molecular systems biology*, **6**(1), 400.

Vreeken, J., Van Leeuwen, M., and Siebes, A. (2011). Krimp: mining itemsets that compress. *Data Mining and Knowledge Discovery*, **23**(1), 169–214.

Vuong, C. K., Black, D. L., and Zheng, S. (2016). The neurogenetics of alternative splicing. *Nature Reviews Neuroscience*, **17**(5), 265–281.

Wahba, G. *et al.* (1999). Support vector machines, reproducing kernel hilbert spaces and the randomized GACV. *Advances in Kernel Methods-Support Vector Learning*, **6**, 69–87.

Wang, Y., Chen, X., Liu, Z.-P., Huang, Q., Wang, Y., Xu, D., Zhang, X.-S., Chen, R., and Chen, L. (2013). De novo prediction of RNA–protein interactions from sequence information. *Molecular BioSystems*, **9**(1), 133–142.

Wethmar, K., Bégay, V., Smink, J. J., Zaragoza, K., Wiesenthal, V., Dörken, B., Calkhoven, C. F., and Leutz, A. (2010). C/EBP$\beta\Delta$uORF mice—a genetic model for uORF-mediated translational control in mammals. *Genes & development*, **24**(1), 15–20.

Wickramasinghe, V. O. and Laskey, R. A. (2015). Control of mammalian gene expression by selective mRNA export. *Nature Reviews Molecular Cell Biology*, **16**(7), 431–442.

Wickramasinghe, V. O., Andrews, R., Ellis, P., Langford, C., Gurdon, J. B., Stewart, M., Venkitaraman, A. R., and Laskey, R. A. (2014). Selective nuclear export of specific classes of mRNA from mammalian nuclei is promoted by GANP. *Nucleic acids research*, **42**(8), 5059–5071.

Xue, S. and Barna, M. (2012). Specialized ribosomes: a new frontier in gene regulation and organismal biology. *Nature reviews Molecular cell biology*, **13**(6), 355–369.

Xue, S., Tian, S., Fujii, K., Kladwang, W., Das, R., and Barna, M. (2015). RNA regulons in Hox 5 [prime] UTRs confer ribosome specificity to gene regulation. *Nature*, **517**(7532), 33–38.

Yan, X., Cheng, H., Han, J., and Xin, D. (2005). Summarizing itemset patterns: a profile-based approach. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 314–323. ACM.

Yang, Y.-C. T., Di, C., Hu, B., Zhou, M., Liu, Y., Song, N., Li, Y., Umetsu, J., and Lu, Z. J. (2015). CLIPdb: a CLIP-seq database for protein-RNA interactions. *BMC genomics*, **16**(1), 1.

Yates, A., Akanni, W., Amode, M. R., Barrell, D., Billis, K., Carvalho-Silva, D., Cummins, C., Clapham, P., Fitzgerald, S., Gil, L., *et al.* (2016). Ensembl 2016. *Nucleic acids research*, **44**(D1), D710–D716.

Yu, G., Li, F., Qin, Y., Bo, X., Wu, Y., and Wang, S. (2010). GOSemSim: an R package for measuring semantic similarity among GO terms and gene products. *Bioinformatics*, **26**(7), 976–978.

Zarnegar, B. J., Flynn, R. A., Shen, Y., Do, B. T., Chang, H. Y., and Khavari, P. A. (2016). irCLIP platform for efficient characterization of protein-RNA interactions. *Nature methods*.

Zhang, S., Wang, W., Ford, J., and Makedon, F. (2006). Learning from incomplete ratings using non-negative matrix factorization. In *SDM*, volume 6, pages 548–552. SIAM.

Zhang, Y., Xie, S., Xu, H., and Qu, L. (2015). CLIP: viewing the RNA world from an RNA-protein interactome perspective. *Science China Life Sciences*, **58**(1), 75–88.