FACULTY OF SCIENCES

FACULTY OF BIOSCIENCE ENGINEERING

FACULTY OF ENGINEERING AND ARCHITECTURE

PREDICTION OF RNA POLYMERASE-DNA INTERACTIONS IN ESCHERICHIA COLI

Word count: 33019

Martin Misonne
Student number: 01607482

Supervisor(s): Prof. Dr. Willem Waegeman
Tutor: Ir. Jim Clauwaert

A dissertation submitted to Ghent University in partial fulfilment of the requirements for the degree of Master of Science in Bioinformatics: Bioscience Engineering

Academic Year: 2017- 2018

GHENT UNIVERSITY

PREFACE

This master thesis marks the very end of my educational journey, which greatly changed direction two years ago. I had taken the decision to follow a different master in one of the multiple bioscience engineering tracks but decided in the end to specialize in bioinformatics. Indeed, I could not resist to switch when I was made to discover this path, and I must say it has been a real revelation.

As a matter of fact, there was a significant challenge though for me in this journey, which was to master the basis of machine learning, an area key to the last evolutions in the area of bioinformatics. A crucial moment in this journey has been the examination for the course of predictive modelling, which required me to participate to a machine learning competition. This at first intimidated me, however, taking part to this experience really made me discover a new passion for big data: this is the reason why I decided to actually choose machine learning as basis theme for my thesis.

But this work would not have been the same without the support of various individuals. First of all, I would therefore like to sincerely thank Jim, the tutor of this thesis, for his continuous support throughout this year: he acted as a real sounding board, and was key to orientate this research in the most relevant direction.

Secondly, I would like to thank Willem, the promoter of this thesis, for the very insightful advice he has been providing me during our review sessions. It has definitely allowed me to get better insights into this problematic.

Thirdly, I would like to thank my family and more specifically my mother and Louise, for the psychological support: it has been a true help during the more difficult moments.

LIST OF ABBREVIATIONS AND SYMBOLS

μ	Growth rate
A	Adenine
AA	Amino acid
A-site	Acceptor site
AUROC (AUC)	Area under the receiver operating characteristic curve
C	Cytosine
ChIP-chip	Chromatin immunoprecipitation and microarray
ChIP-seq	Chromatin immunoprecipitation and sequencings
dA	Deoxy-adenosine
DNA	Deoxyribonucleic acid
E	Ribonucleic acid polymerase core enzyme
EF	Elongation factor
EqEl (*)	Equal Elements string kernel (* improved version)
E-site	Exit-site
Exp.	Exponential growth phase
$E\sigma^x$	Ribonucleic acid polymerase holoenzyme
FN	False negatives
FNR	False negative rate
FP	False positives
FPR	False positive rate
G	Guanine
IF	Initiation factor
LR	Logistic regression
mRNA	Messenger ribonucleic acid
nt	Nucleotide
PCA	Principal components analysis
ppGpp	Guanine tetraphosphate
pRNA	ribonucleic acid product
P-site	Peptidyl-site
PWM	Position weight matrix
qRT-PCR	Quantitative real-time polymerase chain reaction
RNA	Ribonucleic acid

RNAP	Ribonucleic acid polymerase
ROC	Receiver Operating Characteristic
RP	RNA polymerase-promoter complex
RP_c	Closed conformation of the RNA polymerase-promoter complex
RP_o	Open conformation of the RNA polymerase-promoter complex
rU	Uridine
S	Svedberg
Stat	Stationary growth phase
SVC	Support vector classifier
SVM	Support vector machine
T	Thymine
TF	Transcription factor
TFBS	Transcription factor binding site
TN	True negatives
TNR	True negative rate
TP	True positive
TPR	True positive rate
tRNA	Transfer ribonucleic acid
t-SNE	t-distributed stochastic neighbor embedding
TSS	Transcription start site
U	Uracil
UBS	Upstream binding site
WDS (*)	Weighted degree kernel with shifts (* improved version)
XD	X-Dimensional where (X represents a number)
σ^{19}	*fecI*
σ^{24}	*rpoE*
σ^{28}	*fliA*
σ^{32}	*rpoH*
σ^{38}	*rpoS*
σ^{54}	*rpoN*
σ^{70}	*rpoD*
σ^{x}	A σ factor

ABSTRACT

In this thesis, we analyze whether machine learning can be used to bypass part of the laboratory work required to determine interactions between RNA polymerase and DNA. Regulatory networks allow to determine the effects of changing conditions and to connect perturbations in the genome, like mutations, to their downstream or upstream effect. The construction of those networks first requires the interactions between transcription factors and regulatory DNA regions to be determined. Beside this, genetic engineering often implies to insert new genes in a microorganism. Those genes must be recognized by the transcription machinery of the cell to synthetize the protein for which they code. Hence, the choice of the promoters that are incorporated together with those coding elements is of major importance. Testing for such interactions should be made possible using mathematical approaches instead of long and costly laboratory processes.

In this thesis, we will use logistic regression (LR) models and support vector machines (SVM) together with different string kernels to analyze whether modelling those interactions is possible given the dataset at hand. Stacked models will also be employed as they have proven to outperform single model approaches. The problematic will be split in two parts. In the first part, we will analyze the performance of the models for identifying promoters from a set of sequences. Then, we will analyze whether the phase during which a promoter is active can be determined on basis of its sequence. Finally, we will push the problematic up to assigning σ factors (σ^{70}, σ^{38}, σ^{32}, σ^{54} and σ^{28}) that interact with the promoters in general or during a specific growth phase. In the second part, we will propose two classification approaches that combine predictions of two models. We will analyze whether the combination of the models increases the reliability on the top predictions as compared to a single model.

The results showed that promoters can be effectively identified from a set of DNA sequences (0.85 AUC). However, when accounting only for promoter sequences, the performances for assigning the activity during which a promoter is active are 0.72 and 0.58 AUC for the exponential and the stationary phase respectively. Considering the assignment of σ factors to promoters, the average performances are 0.62 and 0.56 AUC for the exponential and the stationary phase respectively. The combination of the models increases reliability of top predictions as compared to the single model. The precision of the top predictions is on average better by 22% and 7% for selecting the promoters that interact with a certain σ factor for the exponential phase and the stationary phase respectively. However, the precision across all the interactions in the top 10 predictions is never completely correct.

In conclusion, identifying promoters in *E. coli* based on the sequence can be effectively performed with our model. However, we were not able to solve the problematic of assigning σ factors to promoters as expected. Nonetheless, we believe that our models would have resulted in better performances on a different dataset. Those models may have a great impact in genetic engineering and for the construction of transcriptional regulatory networks. However, this should be confirmed on another dataset.

CHAPTER 1: INTRODUCTION

1.1. General overview

Microorganisms must deploy a subset of their genetic arsenal at the right place and time if they want to survive. Indeed, part of their molecular tools may be harmful for the cell if they are deployed when they are not required, i.e. under the inappropriate extra- or intra-cellular conditions, and in the right quantity. If they do not react fast enough, it could lead to their death. To make things even more complicated, the cell machinery that allows them to build those tools is limited. For those reasons, the regulation of the expression of their genes must be properly configured.

The regulation of the genes can be represented by a regulatory network. The knowledge of the regulatory network of a given organism allows to determine the effect of perturbations, which can be due to the conditions of growth, mutations, … It has also applications in synthetic biology, industrial biotechnology, healthcare industry and ecology.

The expression of genes is initially regulated at the transcriptional level by transcription factors and, more particularly, by a set of σ factors in *E. coli*. Binding between a σ factor and DNA in the upstream region of a gene, the promoter, is required to express that gene. Each type of σ factor shows different specificity towards DNA sequences. The construction of regulatory networks initially requires determining such interactions. Those relations between DNA sequences and σ factors are determined experimentally. Experimental testing for interactions between all the possible sequences and each σ factor for each organism is a time consuming and expensive effort.

Machine learning can be used to create models that predict interactions without requiring tedious laboratory work. However, those models still require training on labeled data. This data describing interactions between DNA sequences and σ factors must be produced experimentally.

The models can also be used to determine prototype sequences that would bind to any σ factor. Hence, by creating a synthetic promoter with such a sequence, we could maximize the activity of the downstream gene of interest. Or, it can also be used to determine the sequences that bind only with a subset of those σ factors, under a given growth phase.

1.2. Protein synthesis

In this section, we will explain the biological background behind protein synthesis, including transcription and translation. After that, we will see how the synthesis of proteins is regulated at the transcriptional level. The focus will be put on transcription and regulation of transcription, as they are at the core of this master thesis.

1.2.1. General overview of protein synthesis

Proteins are essential for each living organism. They are involved in nearly all processes of life, such as catalysis of the biochemical reactions happening inside and outside cells, cellular transport, intercellular communication and intercellular recognition. They also play a role in cellular structure and immunity. Proteins, also called (poly)peptides, are made of a chain of

amino acids (AA) linked by peptide bonds. There are twenty different AAs. The order of the different AA in the chain corresponds to the primary structure of the protein. The order and composition of amino acids determines the structure of the molecule and its function. The three-dimensional structure of the protein results mainly from hydrogen bonds between amino acids inside the polypeptide. Secondary structure is the protein structure that results from interactions between AAs close to each other. Tertiary structure is the protein structure that results from interactions between more distant AAs. The function of proteins comes directly from the structure. Note that quaternary structure also exists and results from interactions between different proteins. Several protein modifications such as methylation, phosphorylation, acetylation, glycosylation, acylation and cleavage confer them new capabilities, but also influence their activity and their structure (Berg *et al*, 2012).

Deoxyribonucleic acid (DNA) is a linear polymer made up of a chain of nucleotides (nt). Each nucleotide consists of a deoxyribose molecule with a base and a phosphate. The chain of nucleotides is arranged in a backbone of alternating phosphate-deoxyribose groups from which bases protrude. The base is attached on the 1' carbon of the deoxyribose, whereas phosphate groups are attached on 3' and 5' carbons. Thus, the DNA backbone has a 5' end and a 3' end (Figure 1). There are four possible bases that can be attached to the backbone: adenine (A), cytosine (C), guanine (G) and thymine (T). The arrangement of the four different nucleotides inside the DNA chain produces a sequence that is of major importance. Indeed, the primary structure of proteins is encoded in the DNA sequence and, more specifically, in genes. In fact, parts of the DNA are not coding for proteins. There are coding regions that are called genes, and non-coding ones. Non-coding regions can be regulatory sequences. They regulate the process allowing production of a protein, starting from a gene, by interacting with other molecules.

Two strands of DNA interact by complementarity of their bases. Adenine pairs with thymine with two hydrogen bonds and cytosine pairs with guanine with three hydrogen bonds. The hydrogen bonds of a base pair occur between an atom of hydrogen of one base and an atom of oxygen or nitrogen of the other base. The interaction between both strands forms a double helix of DNA. Both strands are oriented in the opposite direction. Thus, the 5' end of one strand matches the 3' end of the other strand (Figure 1) (Berg *et al*, 2012).

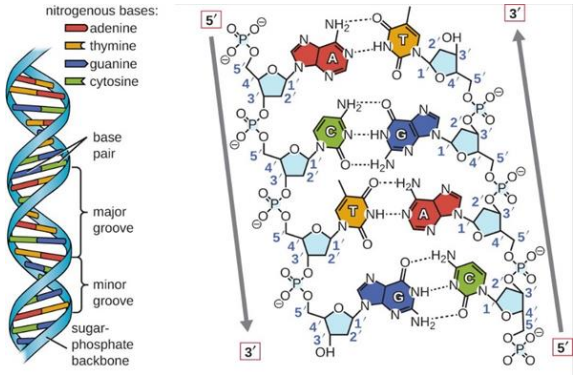


Figure 1. Double helix model. Left: general structure of the DNA double helix. Right: molecular structure of the double helix. The deoxyribose-phosphate backbone is indicated in blue with its bases attached. The dashed lines between bases of the two strands represent hydrogen bonds. (Structure and Function of DNA, 2016)

The first step required for the synthesis of proteins is transcription (Figure 2). Transcription allows for the generation of a ribonucleic acid (RNA) strand that is used to carry the genetic information, called the messenger RNA (mRNA). The mRNA that is synthetized is copied from a DNA template (Kapanidis *et al*, 2006). RNA differs from DNA by the sugar that is used for its backbone and by the possible bases it is composed of. The backbone's sugar is not deoxyribose but ribose and adenine is paired to uracil (U) instead of thymine. Moreover, RNA is generally single-stranded but double-stranded (ds) RNA also exists (Monsion *et al*, 2018). The DNA template is copied into a mRNA by base pairing. Thus, the mRNA sequence is complementary to its template, and the 3' end of the template matches the 5' end of the mRNA. The template strand is always read in the 3' - 5' direction. The complementary strand is called "coding" or "sense" strand as its sequence is the same as the synthesized mRNA, with T replaced by U. A positive and a negative DNA strand is arbitrarily defined in *E. coli*. Both the positive and the negative strand can be used as a template for transcription, but only one is used for a given gene (Weaver, 2011; Berg *et al*, 2012).

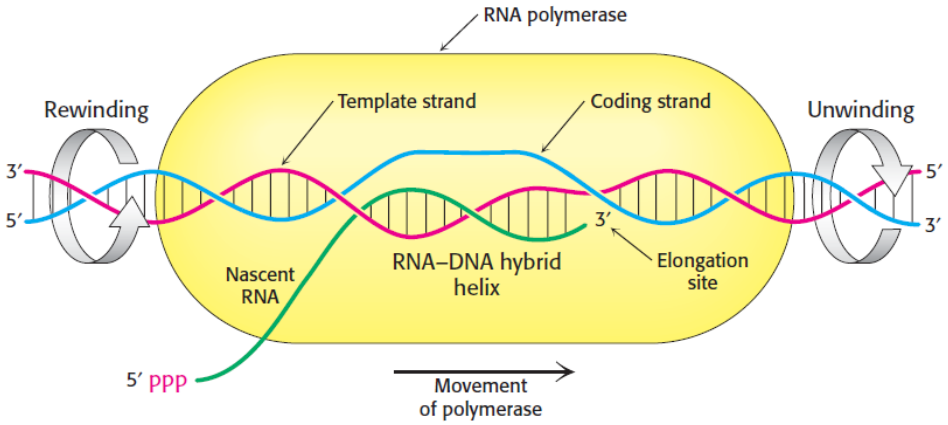


Figure 2. Transcription. The synthetized RNA strand is indicated in green. The 3' end of the RNA forms an RNA-DNA hybrid with the template strand with which it pairs. The RNA has the same sequence as the DNA coding strand indicated in blue, except that T is replaced by U. (Berg *et al*, 2012)

The mRNA sequence is then translated into a polypeptide chain based on the genetic code (Table 1) during a process called translation. The genetic code allows to match a RNA- (or a DNA-) sequence of three nucleotides called codon, with a given amino acid. The genetic code is valid for all living organisms. Translation is the step during which the synthesis of the protein occurs (Weaver, 2011; Berg *et al*, 2012).

Table 1. The genetic code. The codons are shown at the upper side of the table. The 3- letter codes and 1-letter codes at the bottom of the table represent the different AAs. UAA, UAG and UGA correspond to a stop codon. (Alberts *et al*, 2015)

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

Translation

In prokaryotes, translation and transcription are not spatially separated in the cell, both occur in the cytosol. Ribosomes are molecular complexes composed out of ribosomal RNA (rRNA) and proteins. They perform translation by reading the mRNA strand from its 5' end to its 3' end. The codons of the mRNA are read one after the other, and the corresponding AA-chain is polymerized. The entire mRNA sequence is not always used to code for an AA. Indeed, there may be a conserved and untranslated region, AGGAGG, at the 5' end called the Shine-Dalgarno box. This sequence allows base-pairing with an rRNA of the ribosome on the ribosome binding site, located ~8 bp upstream of the start codon: AUG (or GUG). This interaction allows the alignment of the ribosome with the start codon (Alberts *et al*, 2015; Weaver, 2011; Berg *et al*, 2012).

Adding of a free AA to the peptide chain is not thermodynamically favorable. AA-esters, called activated AAs, are necessary to allow peptide bond formation during polymerization of the peptide chain. These are carried to the mRNA by a transfer RNA (tRNA), forming the aminoacyl-tRNA or charged tRNA. A codon of the mRNA binds the appropriate aminoacyl-tRNA by complementarity with a sequence on tRNA called anticodon. Amino acids are added to the appropriate tRNA by an enzyme called aminoacyl-tRNA synthetase. There is at least one specific enzyme for each AA (Alberts *et al*, 2015; Weaver, 2011; Berg *et al*, 2012).

Ribosomes are made out of a small subunit and a large subunit, 30S and 50S respectively. During initiation of translation, the small subunit, containing three sites: E (exit), P (peptidyl) and A (aminoacyl), binds to the mRNA on the ribosome binding site, helped by three initiation factors (IF): IF-1, -2 and -3. Simultaneously, a tRNA carrying the AA matching the start codon enters the P site. The IFs are then released from the 50S subunit to bind to the complex and form the complete ribosome (70S) (Alberts *et al*, 2015; Weaver, 2011; Berg *et al*, 2012).

Then starts the elongation, carried by three elongation factors (EF). A tRNA goes into the A-site and stays only if its anticodon complements the mRNA codon on the A-site. The peptide (now one AA) linked to the tRNA at the P-site is transferred to the AA of the tRNA at the A-site and a peptide bond is formed. This reaction is catalyzed by the ribosome. Then, the large subunit of the ribosome translocates in the 3' direction. This displacement moves the tRNA from the A-site to the P-site and the tRNA at the P-site to the E-site. The tRNA at the exit site leaves the ribosome and the small subunit of the ribosome translocates under the large subunit. Then, the cycle restarts (a new tRNA enters the A-site, …). Figure 3 shows the steps of addition of the fourth AA (Alberts *et al*, 2015; Weaver, 2011; Berg *et al*, 2012).

Elongation is stopped when a stop codon (UAA, UAG, UGA) comes into the A-site. Indeed, a stop codon is recognized by one of the release factors: RF-1 or RF-2. This induces releasing of the polypeptide. Subsequently, the interaction between RF-3 and the A-site provokes detachment of the ribosome subunits (Alberts *et al*, 2015; Weaver, 2011; Berg *et al*, 2012).

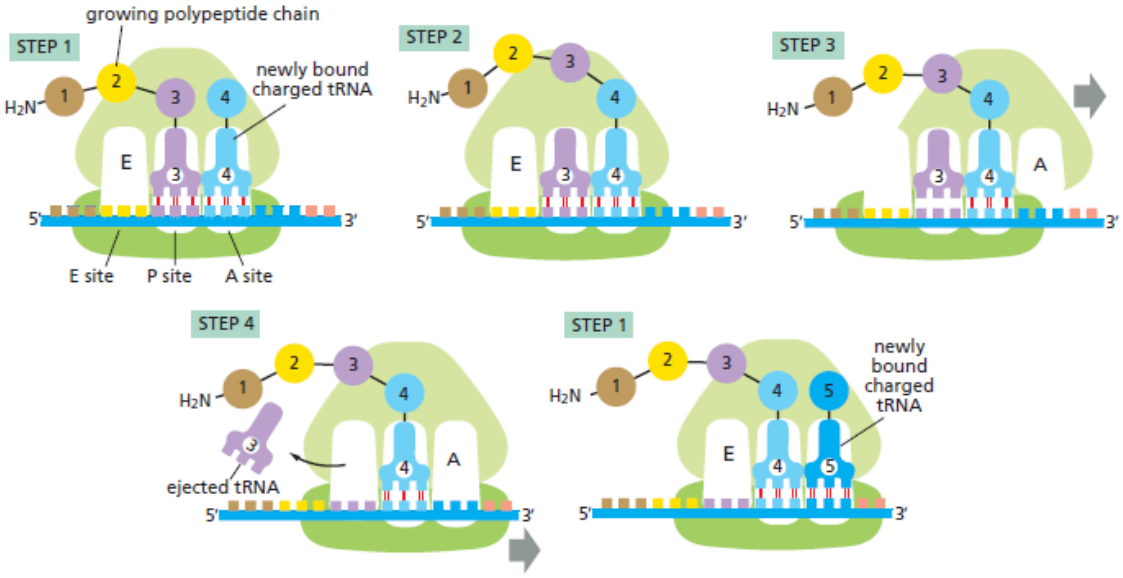


Figure 3. Translation of an mRNA. The mRNA is represented by the blue line and the ribosome is shown in green. The light green part corresponds to the 50S subunit and the dark green part correspond to the 30S subunit. The polypeptide that is synthetized is indicated by the colored circles. (Alberts *et al*, 2015)

1.2.2. Transcription in *E. coli*

Transcription is discussed thoroughly in this subsection, apart from the general overview on protein synthesis. First, we will describe the molecular complex responsible for this process. Then, we will go through the three steps composing this biological process.

RNA polymerase

The molecular complex responsible for transcription is a multi-subunit RNA polymerase composed of four types of subunits: α (two copies) , β, β'and ω, forming the RNA polymerase core enzyme (E) (Finn *et al*, 2000).

Both α subunits (329 AAs in *E. coli*) are made out of two independently folding domains that are joined together by a linker of ~20 AAs (Browning & Busby, 2004). The amino-terminal domains (αNTD, AA 1-235) allow the assembly of β and β' subunits after dimerization. The carboxy-terminal domains (αCTD, AA 250-329) can bind DNA (Browning & Busby, 2004). β and β' subunits represent the active site cleft of the complex which contains two divalent metal ions (usually Mg^{2+}), playing the role of electron withdrawer. β and β' are the biggest subunits of the complex, they are made of 1342 and 1407 residues respectively in *E. coli* (Browning & Busby, 2004). All bacterial β' subunits seem to contain the AA-string NADFDGD from which aspartate residues chelate both divalent metal ions (Weaver, 2011). Both β and β' subunits are responsible for binding: to double-stranded DNA downstream of the synthesis direction, to DNA-RNA hybrids during transcription and to RNA (Murakami, 2015). The small ω subunit (91 AAs) helps in the last step of the assembly of the RNA polymerase core enzyme. That is, the association between β' and $\alpha_2\beta$. Nevertheless, a study of Gunnelius *et al* showed in 2014 that the ω subunit is not essential in *E. coli* (Browning & Busby, 2004; Gunnelius *et al*, 2014).

The RNA polymerase core enzyme requires a transcription factor to interact with DNA and initiate transcription. This transcription factor is a σ factor. Together, they form the RNA polymerase holoenzyme ($E\sigma^x$) (Gunnelius *et al*, 2014). σ factors allow principally to: associate the $E\sigma^x$-promoter complex (RP) in the initially "closed" conformation (RP_c), stabilize the

complex in "open" conformation (RP_o) and interact with transcription activators for RNA synthesis (Paget, 2015).

σ factors are proteins containing up to four different domains interacting with the core enzyme (Nagai & Shimamoto, 1997). σ_2, σ_3 and σ_4 specifically interact with promoter elements. $\sigma_{1.1}$ occupies the active site in RP_c before double stranded DNA in RP_o. Note that this domain is absent from most of the σ factors and can inhibit or promote transcription (Paget, 2015; Vuthoori *et al*, 2001). Thus, the σ_1 domain needs to be moved away from the RNA polymerase cleft to form the active RP_o (Murakami, 2015). Seven types of σ factors are known in *E. coli*; a house-keeping σ factor: σ^{70} and six alternative minor σ factors: σ^{54}, σ^{38}, σ^{32}, σ^{28}, σ^{24} and σ^{19} (respectively: *rpoD, rpoN, rpoS, rpoH, fliA, rpoE* and *fecI*) (Shimada *et al*, 2017; Cho *et al*, 2014). The group of alternative σ factors mainly regulates expression of genes involved in the response to environmental stress-conditions, but also in auxiliary processes such as nitrogen fixation or flagellar assembly (Glyde *et al*, 2017). The polymerase and σ factors are limiting elements for the transcription of genes. The competition between σ factors for the RNA polymerase and the competition between promoters allow to regulate gene expression in the cell (Browning & Busby, 2004; Maeda, 2000).

Anti-σ factors have an antagonist role to σ factors. Some of those molecules prevent the association of σ factors with the core enzyme by interacting principally with the alternative σ factors, less with σ^{70}. As σ factors are limited in the cell, transcription of genes that are dependent to the targeted σ factor is diminished. Other anti-σ factors bind to σ factors but they still allow the interaction with the core enzyme. For example, AsiA binds with the σ_4 domain, preventing the initiation of the transcription. This might be caused by the impossibility of the σ_4 to interact with the -35 region of the promoter (see Transcription initiation), or by preventing the association of σ_4 with the β subunit of the core enzyme (Dove *et al*, 2003).

Transcription initiation

Here, we will discuss the first step of the transcription process. Locations indicated with numbers refer to the relative position towards the transcription start site, indicated by 0, $-x$ refers to the x^{th} position upstream the TSS.

Studies performed on *Thermus aquaticus* and *Thermus thermophilus* revealed that promoter region -41 to -7 stands outside the RNA polymerase active site cleft, at the surface of the complex. This place corresponds to where the σ factor is located. This observation has shown that it is the σ factor that interacts with the promoter rather than the RNA polymerase itself (Murakami, 2015).

Promoters contain four sequence elements involved in the interaction with the σ factor of the RNA polymerase. Those elements are specific to each σ factor. In *E. coli*, for σ^{70}, two of them are hexamers with consensus sequences TTGACA and TATAAT and are located respectively around positions -35 and -10. Those hexamers are the principal elements responsible for the promoter recognition by σ_4 (-35 box, by subregion $\sigma_{4.2}$) and σ_2 (-10 box, by subregion $\sigma_{2.4}$) domains of the RNA polymerase (Browning & Busby, 2004). Considering σ^{54}, interaction between promoter and the σ factor occurs at consensus sequences [CT]TGGCA[CT][GA] and TGC[AT][TA] around regions -24 and -12 respectively. Transition to RP_o conformation depends

on enhancer-binding proteins (Lin *et al*, 2014). Another element is the extended -10 element, which is found upstream the -10 box and consists of a 3-4-mer: TGn (Sanderson *et al*, 2003). This pattern is recognized by the σ_3 domain, is present in ~20% of the *E. coli* promoters and can trigger promoter activity up to more than 100-fold (Sanderson *et al*, 2003; Ross *et al*, 2001). The last element is a ~20 bp sequence located upstream the -35 element up to ~-90 (Browning & Busby, 2004; Saecker *et al*, 2011). This region is called the "UP element" and is not recognized by the σ factor but rather by the αCTDs of the RNA polymerase (Figure 4). Promoters that do not contain an UP element are called core promoters (Browning & Busby, 2004; Ross *et al*, 2001; Gourse *et al*, 2000).

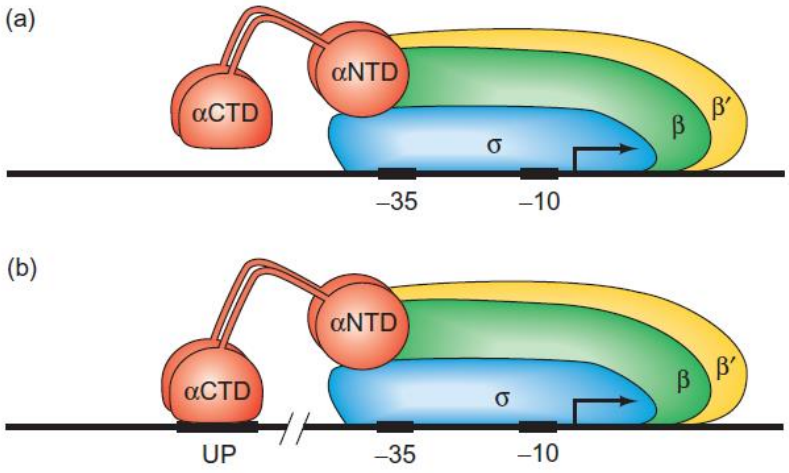


Figure 4. Model for the function of the C-terminal domain (CTD) of the polymerase α-subunit. (a) In a core promoter, the α-CTDs do not interact with the promoter. (b) in a promoter with an UP element, the α-CTDs interact with the UP element. The dark line represents the promoter. The colored shapes over the promoter represent the subunits of the RNA polymerase holoenzyme. (Weaver, 2011)

A consensus sequence is a sequence that is over-represented across promoters but is not the "optimal" sequence. It also differs for each σ factor in *E. coli*. The closer a promoter sequence is from the consensus sequence, the more affinity it will have for the σ factor and thus express the gene more efficiently (Browning & Busby, 2004). However, most of the promoters do not have a consensus sequence. Hence, an equilibrium is set between the activities of the promoters (Browning & Busby, 2004).

For the transcription to start, the RNA polymerase holoenzyme-promoter complex (RP) must switch from the inactive closed conformation to the active open conformation. This shift allows to initiate transcription at the transcription start site (TSS) and not in the promoter region (Glyde *et al*, 2017).

The open conformation of the promoter is obtained by a process called "isomerization" and results in an unstable open complex (Saecker *et al*, 2011; Browning & Busby, 2004). Isomerization allows to separate both DNA strands around position -13 to +2, forming a transcription "bubble" from which position +1 of the template strand is placed in the active site. This displacement of the template stabilizes the open complex (Saecker *et al*, 2011). Three aromatic AAs (Phe248, Tyr253 and Trp256) from the σ factor might be responsible for the isomerization. These AAs are well conserved across organisms. Isomerization might occur by linkage of those AAs to the non-template strand around the -10 box (Weaver, 2011). The

non-template strand or coding strand still interacts with the RNA polymerase and regulates the formation of the complex, its lifetime, the selection of the transcription start site and abortive synthesis (Nandy Mazumdar *et al*, 2016).

Indeed, the RNA polymerase-DNA complex can evolve in abortive or productive synthesis (Saecker *et al*, 2011). During synthesis, α-phosphate of the lastly arrived nucleoside triphosphate is covalently linked to the 3'-OH end of the neosynthesized RNA. The α-phosphate corresponds to the phosphate group that is directly linked with the ribose. The RNA chain and the transcription "bubble" grow while the RNA polymerase holoenzyme still interacts with the promoter (Alberts *et al*, 2015). Thus, template DNA is continuously pulled inside the enzyme complex, this is called "scrunching". This process induces stress and competition between two possible paths: releasing the RNA strand to diminish the stress (abortive synthesis) or keeping on extending it (productive synthesis). The size of the aborted RNA chain is a function of the promoter sequence and conditions but is not yet elucidated (Saecker *et al*, 2011). Abortion can happen several times before the productive synthesis to occur (Alberts *et al*, 2015). The stress is automatically avoided at a critical RNA chain size of 11 nucleotides by perturbation of the interaction between the RNA polymerase and the promoter (Saecker *et al*, 2011). Note that initiation does not necessarily end up with the release of the σ factor from the RNA polymerase (Bar-Nahum & Nudler, 2001; Mukhopadhyay *et al*, 2001).

Transcription elongation

The β' subunit of the polymerase contains a valine residue that is in contact with the minor groove of the DNA downstream the transcription bubble. This AA might act by causing the screw-like motion by turning around the minor groove of the DNA. This valine might also stop the DNA to prevent it from entering or escaping the polymerase. (Weaver, 2011)

The action of separating paired nucleic acids is described as helicase activity. During elongation, the polymerase melts DNA downstream (helicase) and unmelts it upstream of the synthesis direction. This results in a transcription bubble covering around 17 bp and containing an RNA-DNA hybrid of ~9bp up to position +1, where the new nucleotide is added. As downstream DNA is double-stranded up to position +2, only position +1 is available for incoming nucleotides. Hence, new nucleotides are added one by one (Vassylyev *et al*, 2007). The arginine 422 of the β subunit interacts with the nucleotide of the template strand at position +1. This AA might thus be implied in the proofreading process. Proofreading consists in verifying that the correct nucleic acid is added. Indeed, the new nucleotide is added to the elongation complex in a "pre-insertion state" which enters an "insertion state" if bases are correctly paired and linked to the appropriate sugar (ribose). The polymerization of the RNA chain is catalyzed by both Mg^{2+} ions in the active site, acting as an electron withdrawer. The nascent RNA goes out from the polymerase by the exit channel. (Weaver, 2011)

What limits the length of the RNA-DNA hybrid is on one hand the size of the transcription bubble and on the other hand a hydrophobic pocket that captures the first RNA base displaced from the hybrid (upstream). Indeed, contrarily to DNA polymerase, RNA polymerase must unhybridize the synthesized RNA and DNA template to rewind DNA upstream of the transcription bubble (Jiang *et al*, 2004).

Elongation brings tension in DNA downstream and upstream the transcription bubble. This

stress is released thanks to creation of positive supercoils downstream and negative supercoils upstream of the bubble. Supercoils are then relaxed by the action of topoisomerases. This costs less energy than making the RNA polymerase rotate to follow the DNA's twist during the elongation, which would also surround the synthetized RNA around the DNA template (Weaver, 2011).

Elongation is not a continuous process, pauses occur and those have an important regulatory role. Pausing allows to coordinate transcription and translation and help the folding of the nascent product. Both steps of transcription permit regulators to bind to the complex and are required for termination of the transcription (Weixlbaumer *et al*, 2013). Pausing can have several causes. The first cause is the addition of the wrong nucleotide. The second cause is the presence of a promoter-like sequence in the non-template strand, interacting with the σ factor, if the latter is still present. The third cause is the formation of an RNA hairpin of ~11 nucleotides at the exit channel of the polymerase (Yakhnin & Babitzke, 2010; Weixlbaumer *et al*, 2013).

When the wrong nucleotide is added, backtracking occurs. This process is carried by proteins GreA and GreB which activate RNase activity of the polymerase (3' to 5' direction). During backtracking, the polymerase goes in the opposite direction, removing completely the 3' end of the RNA from the active site. This movement induces pausing which can also lead to a complete interruption of the transcription. GreB also allows to prevent this pause to occur.
However, these two proteins are not mandatory for proofreading. As the last incorporated nucleotide does not match with the DNA template, it is more flexible. The wrongly incorporated nucleotide can thus bend back to enter in contact with the Mg^{2+} of the active site. The metal ion might be involved in the RNase activity. This cannot occur if the appropriate nucleotide is added as it will not be flexible enough to interact with Mg^{2+}. (Weaver, 2011)

The formation of an RNA hairpin at the exit channel provokes pausing of transcription. Indeed, the RNA hairpin interacts with the β-flap of the β subunit of the polymerase. This interaction might induce displacement from the active site of critical residues for polymerization activity (Kang *et al*, 2018). The NusA protein enhances this effect by stabilizing the interaction between the β-flap and the hairpin. Hairpin-formation is due to apparition of a reverse complemented repeat in the sequence. The repeat is separated by a gap with a "random" sequence, forming the loop of the hairpin (Toulokhonov *et al*, 2001).

Transcription termination

There are two different types of transcription termination signals or "terminators". One type of terminator depends only on the RNA polymerase whereas the other depends on proteins called "rho".

The rho-independent terminator is a T-rich region preceded by a reverse complemented repeat in the non-template strand, which favorizes the formation of a hairpin in the RNA at the exit channel. The T-rich region allows to form weak interactions in the RNA-DNA hybrid (rU-dA interactions) to help dissociation of both strands. The separation of the transcript from DNA causes pausing of transcription. This pause allows the hairpin to form and destabilizes the elongation complex. Because of this instability, the transcript further detaches from the template. RNA escapes from the polymerase and the transcription bubble can rewind (Figure

5, left). Thus, hairpin-formation facilitates termination but is not mandatory.
The hairpin-forming sequence can bind a region of the core polymerase called the upstream binding site (UBS). This association inhibits hairpin-formation and thus termination. NusA avoids inhibition of termination by weakening the interaction between the transcript and the UBS. (Yarnell & Roberts, 1999; Farnham & Platt, 1980)

Rho-dependent termination depends on proteins called “rho” and a reverse complemented repeat. Contrarily to rho-independent termination, this process does not require a U-rich region at the end of the transcript. Rho is a doughnut shaped hexamer made out of the same subunits. This protein initially interacts with the polymerase at the beginning of transcription. When the transcript is long enough, rho binds RNA polymerase on a sequence called the “rho loading site”. This interaction creates an RNA loop between the rho binding site and the RNA escaping from the polymerase. During transcription, the nascent RNA is fed through the center of the doughnut shaped protein. The RNA loop remains as rho cannot catch up the RNA continuously flowing out of the polymerase. At the end of transcription, a hairpin is formed because of the reverse complemented repeat. The hairpin causes the transcription to pause, allowing rho to catch up with the lastly synthesized ribonucleotides. RNA synthesis gets blocked because the RNA loop gets tightened, the elongation complex is “trapped” (Figure 5, right). Then, rho dissociates the RNA-DNA hybrid with its helicase activity. The transcript is released, ending the transcription (Brennan *et al*, 1987; Roberts, 1969; Epshtein *et al*, 2010) .

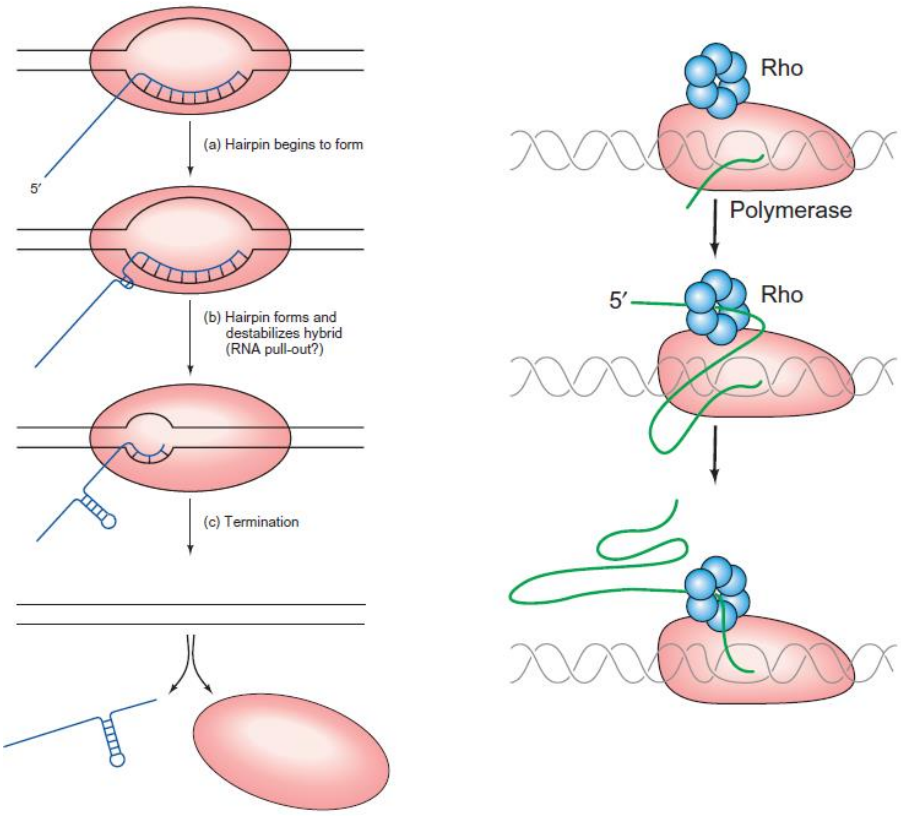


Figure 5. Rho-independent termination (left) and rho dependent termination (right). The RNA polymerase is represented by the pink shape. The rho-protein is represented by the blue hexamer. At the last step of the rho-dependent termination, we see that the RNA loop gets tightened, inducing the release of the mRNA. (Weaver, 2011)

1.2.3. Regulation of transcription is mediated by σ factors

In this subsection, we will discuss the principles and importance of the regulation of protein synthesis, and more particularly the regulation of transcription. Therefore, we will first explain the differences between the growth phases of microorganisms. Then, we will explain how this regulation works at the molecular level. The latter part is of major importance considering the subject of this master thesis. Indeed, interactions between σ factors and promoters change while conditions change and those molecules explain part of the regulation of transcription.

Growth phases

The growth of a microorganism in nature can be studied thanks to its behavior in a batch fermentation. Fermentations are used in the bioindustry to produce complex molecules by cultivation of microorganisms in the presence a substrate, which they feed on. Typical examples of molecules produced by fermentation are: bioethanol (Rolfe *et al*, 2012), lactic acid (Reddy *et al*, 2008), vitamin B12 (Keuth & Bisping, 1994), penicillin (San & Stephanopoulos, 1989) and coenzyme Q_{10} (Kien *et al*, 2010). The process occurs in bioreactors under controlled conditions. During batch cultivation, a certain quantity of substrate is added to an empty reactor. Then, microorganisms are inoculated to start the fermentation.

There are five distinct growth phases (Figure 6): the lag phase, the exponential phase, the stationary phase, the death phase and the long-term stationary phase. The fermentation first starts by the lag phase. The microorganism needs to adapt to the new environment: pH, temperature, substrate, … It does not multiply yet as it requires a new set of molecules to cope with the new conditions. After a time of adaptation, the cells start to grow and multiply while consuming the substrate. This phase is called the log phase or exponential phase as the number of cells grows exponentially. Then, two factors can cause the organism to enter in the stationary phase. This phase corresponds to the stop of replication of the cells. An essential nutrient can become limiting or a toxic product can accumulate. When the cells run out of their energy reserves, they enter the death phase. Microorganisms die and their number decreases exponentially with time. During the long-term stationary phase, dead cells release nutrients that are used by survivors, which can multiply. This is called cross-feeding. There is an alternating increase and decrease in the number of living cells. (Rolfe *et al*, 2012; Pletnev *et al*, 2015)

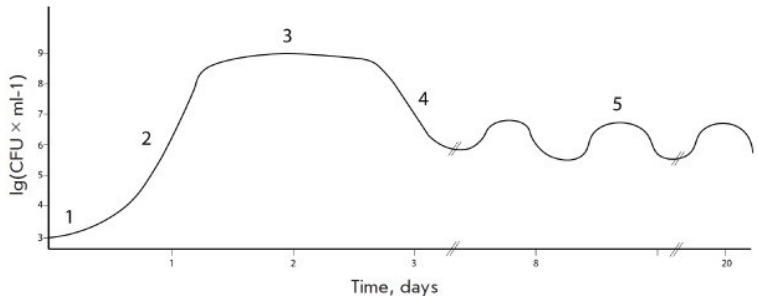


Figure 6. Bacterial growth curve showing the five different growth phases. 1: lag phase, 2: exponential phase, 3: stationary phase, 4: death phase and 5: long-term stationary phase. (Pletnev *et al*, 2015)

The transition between the exponential phase and the stationary phase in *E. coli* comes along with reprogramming of the physiology of the cell and gene expression. This transformation is

driven by the modulable selectivity of the different σ factors for the different promoters. This variation is due to a modification of the relative concentrations of each σ factor and to interactions with other molecules (Bernardo *et al*, 2006; Typas *et al*, 2007; Wassarman & Storz, 2000; Kang *et al*, 1997).

1.2.4. Promoter activities and promoter specificity

The limited transcriptional resources must be appropriately distributed across the set of genes of the cell. Depending on the conditions, certain promoters need to be more active than others. To this end, two factors are of major importance: growth rate (μ) and growth conditions. The growth rate corresponds to number of cell divisions occurring per unit of time. This metric generally lies between 0.2 and 1.3 /h (Andersen & Von Meyenburg, 1980). It appears that the general promoter activity of the cell depends only on growth rate. Moreover, this activity follows a power-law distribution. In fact, the distribution derives from a mixture of two log-normal distributions: metabolic promoters and ribosomal promoters (Figure 7). Both distributions remain constant for a given growth rate. However, the different metabolic promoters show different activities in function of the growth conditions (Zaslaver *et al*, 2009).

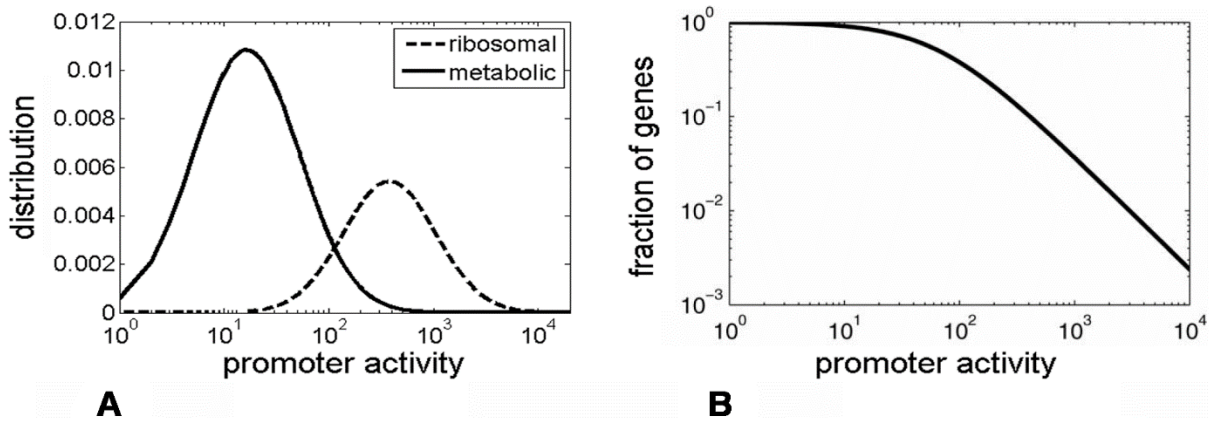


Figure 7. Heavy-tailed distribution obtained by a mixture of two log-normal distributions. (A) Log-normal distributions with the observed mean and standard deviation of ribosomal promoters (dashed) and metabolic promoters (solid line) at $\mu = 0.8$ divisions per hour in glucose medium. The ribosomal function was multiplied by 5 for clarity. **(B)** Rank frequency plot for the resulting mixture of these two 'scale rich' classes. Note: figure and legend were downloaded from the paper of Zaslaver *et al* (2009).

We will now focus on the transition between the exponential phase and the stationary phase. In E. coli, this phase switch implies variation of the growth rate and restructuration of the gene regulatory network (Raffaelle et al, 2005). It is straightforward to think of how genome-wide promoter activity becomes lower. As the nutrient availability is low, the cell cannot afford transcribing all its genes. Thus, the promoter activity at the general level is diminished. However, it is more complex to understand how the global pattern of metabolic promoters' activities is modified. The number of RNAP core enzymes is limited in the cell, creating a competition between the different σ factors. When switching growth phase, the relative concentrations of the σ factors change. Therefore, the pool of RNA polymerase holoenzymes changes, such as the general selectivity for the different promoters (Kang et al, 1997). When switching from the exponential to the stationary phase, the concentration of σ^{38} is increased to one third of the most abundant σ factor, the housekeeping σ factor (σ^{70}) (Typas et al, 2007). Hence, dominance of σ38 should be increased.

However, this switch towards alternative σ factors is not only a matter of σ factor concentrations. Indeed, the molecular characteristics of the RNA polymerase core enzyme also differs. It has been shown that entering the stationary phase implied association of poly-phosphates to the core enzyme (Kang *et al*, 1997). In fact, those molecules correspond to interaction of $E\sigma^{70}$ holoenzymes with guanosine tetraphosphate (ppGpp) alarmone. This nucleotide allows the dissociation of the $E\sigma^{70}$ complex in synergy with another protein: DksA. The core enzyme is thus more available for alternative σ factors (Bernardo *et al*, 2006; Typas *et al*, 2007). ppGpp is an adaptive response to amino acid starvation (Artsimovitch *et al*, 2004; Jishage *et al*, 2002; Perederina *et al*, 2004).
In addition, 6S RNA further reduces the activity of σ^{70}- specific promoters. This RNA of 184 nt competes with promoters by interacting specifically with $E\sigma^{70}$. This conserved molecule among bacteria imitates the secondary structure of a promoter's transcription bubble (Barrick *et al*, 2005). The 6S RNA can also be used as a template for transcription when nutrient availability increases, releasing an RNA product (pRNA). Synthesis of pRNA provokes separation of the 6S RNA-holoenzyme complex. Thus, when conditions become favorable for growth, the activity of σ^{70}-dependent promoters is recovered (Cavanagh *et al*, 2012; Chen *et al*, 2017).
The *E. coli*'s alternative σ factor σ^{38} is the most closely related to σ^{70}. Indeed, they have a similar -10 box (TAtaaT and CTAtacT, for σ^{70} and σ^{38} respectively). There is no distinctive -35 box for σ^{38} (Cho *et al*, 2014). Both facts allow large overlap of the regulated promoters between both σ factors. During the stationary phase, a protein called Crl favorizes $E\sigma^{38}$ interactions (Typas *et al*, 2007). Ironically, this molecule induces also the synthesis of RssB, the protein responsible for the proteolysis of σ^{38}. The σ factor is protected from degradation while it interacts with RNAP ($E\sigma^{38}$). However, the effect of Crl promoting $E\sigma^{38}$ formation is favorized over the effect of σ^{38} proteolysis. Thus, competitivity of this alternative σ factor is increased during stationary phase.

1.3. Machine learning

Nowadays, an enormous quantity of information is generated every day which is mostly accessible online (Jacobson Ralph, 2013). Data is now stored online whereas it was initially saved on a sheet of paper. The large amount of information that is generated needs to be processed to extract tendencies, make conclusions, create knowledge, but also to be able to make predictions. Companies must take decisions that are supported by the data, if they want to grow and minimize the risk of failure. It is nearly impossible for a human to do so. The computational speed of computers allows us to process the data faster and in an automated way. The use of computers for data analysis is applicable for a wide variety of fields: sales, transport, healthcare, agriculture, biology, in banking, telecommunications, etc. (Chen *et al*, 2014a; Assunção *et al*, 2015; Carbonell, 2016).

1.3.1. Machine learning types

A dataset is a table composed of observations. An observation is an entry in the dataset for which different parameters are measured. The measurement of the parameters can be either qualitative or quantitative. Those parameters are used to build a model which will allow to predict an output. A model is a mathematical function that depends on the parameters

provided in the dataset. For some datasets, the output is already present for the observations. In this case, the output allows to train a model to make predictions for new observations. Training or fitting a model corresponds to assigning coefficients to its parameters. This type of machine learning procedure is called supervised learning. Supervised learning makes use of labeled data to train a model. When the observations in the data are not assigned to a label we enter in the field of unsupervised learning.

Supervised learning

Depending on the type of output that the observations are labeled with, the machine learning approach will be different. If the output represents a category (label or class), it is a classification problem. The output is qualitative. If the output is a quantitative value, the problem is referred to as regression (Hastie *et al*, 2009).

Classification is the type of machine learning model used to assign qualitative values (classes) to observations. Examples of classification problems would be: assigning mails to “spam” or “mail” and determining whether a bag undertaking an X-ray control at the airport contains forbidden items or not. These are examples of binary classification problems. The observations must be classified into one of two classes. Parameters used to classify mails could be: the number of misspelled words, the number of words, the presence of hyperlinks, the presence of the words “free”, “win”, etc. Parameters used to report dangerous bags could be: presence of metallic objects, presence of liquids, etc. In those type of problems, the model is required to build a decision function that will allow discriminating observations of both classes. Multiclass classification consists in a problem for which the observations may be assigned to several classes. For instance, the picture of an animal is shown to the model which must determine whether the animal is a deer, a wild boar, a pheasant, or a fox. Only one class from four can be assigned to the image. Multilabel classification is a problem for which several classes can be assigned to an observation. For instance, several animals may be present in a picture. There are multiple labels to be predicted by the model, one label per animal. The model will determine for each possible target whether it is present or not. A common way to solve a multilabel classification problem is to train a separate binary classifier for each possible target.

Regression refers to machine learning models that predict quantitative outputs. For example, a doctor who wants to predict the level of glucose in the blood of its patient on basis of a multitude of parameters: has he eaten recently, what is his hearth rate, etc. An economist who tries to determine tomorrow’s price of a company’s share on the stock market based on the evolution of the last 5 days. Here, the parameters could be: price on day -5, on day -4, … up to day 0 and we want to predict the price on day +1 (tomorrow).

Unsupervised learning

Unsupervised learning allows to build models that classify observations without requiring the explicit labelling of the observations. Observations do not have an associated response. What can be done in such cases is trying to understand the relationships between the parameters or between observations. An example of unsupervised learning approach is clustering. Clustering methods allow to group similar observations in the data. There exist multiple

methods for clustering observations. A well know clustering family that is used in this thesis is hierarchical clustering.

Hierarchical clustering produces a user-defined number of clusters from the observations. Those clusters are nested with each other in a hierarchical manner. They form a tree in a way that the root contains all the observations from the data and each leaf contains a subset of similar observations. In this tree, clusters (leaves) present in the same branch resemble to each other mother than observations present in other branches. For instance, hierarchical clustering can be used to build phylogenetic trees which group organisms based on of the similarity of their 16S rRNA gene sequences (Cai & Sun, 2011).

1.3.2. Machine learning models used

In this subsection, we will present machine learning models that were used in this master thesis. We explain the rationales behind logistic regression and support vector machines.

Logistic regression (LR)

Logistic regression fits a linear model and transforms it with the logistic function. It is used for binary classification problems. A linear model is given by Eq. 1. $x = (x_1, x_2, \ldots, x_p)$ contains the p values assigned to observation x, one per feature. β_0 corresponds to the intercept, that is the response of the equation when all the values for $x = 0$. β correspond to the coefficients vector $(\beta_0, \beta_1, \ldots, \beta_p)$, there is one coefficient per feature.

$$f(x) = \beta_0 + \sum_{i=i}^{p} \beta_i x \tag{1}$$

In binary classification, the label assigned can be either 1 (positive class) or 0 (negative class). The output of the function described in Eq. 1 is a continuous value. We can assign the label 0 or 1 to an observation on basis of the output of the function by using a threshold. Using a cutoff of 0.5, an observation will be classified as positive if the output is bigger than 0.5. The response of the function can be seen as the probability for an observation to belong to class 1. The problem is that the output of this function can be outside [0, 1], which is not possible for probabilities. Applying a logistic function to this linear model keeps the output between the desired boundaries and forms Eq. 2. $p(x)$ is the probability for observation x to belong to class 1.

$$p(x) = \frac{e^{\beta_0 + \sum_{i=i}^{p} \beta_i x}}{1 + e^{\beta_0 + \sum_{i=i}^{p} \beta_i x}} = \frac{1}{1 + e^{-(\beta_0 + \sum_{i=i}^{p} \beta_i x)}} \tag{2}$$

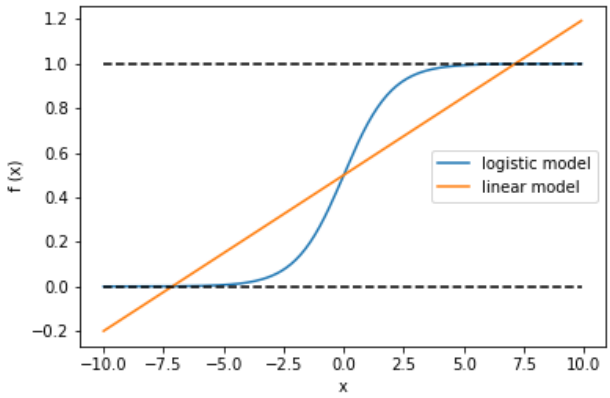


Figure 8. Transformation of a linear model into a logistic model. The linear model $\boldsymbol{f(x) = 0.5 + 0.07x}$ is transformed into the logistic model $\boldsymbol{p(x) = \frac{1}{1+e^{-(0.05+0.07x)}}}$. The output of the logistic model is maintained between [0, 1] whereas the output of the linear model can be outside [0,1].

Support vector classifier (SVC)

SVC fits a linear hyperplane to separate classes (James *et al*, 2000). The hyperplane that is fit is the one that lies the farthest possible from the observations of each class. For this reason, the classifier is also referred to as the maximal margin classifier (Figure 9). The margin is the perpendicular distance between the hyperplane and the observations that lie the closest from it. Those observations are called support vectors, they are the only one influencing the shape of the hyperplane. The support observations can lie either: on the margin, between the hyperplane and the margin or at the wrong side of the hyperplane. Indeed, it is not always possible to separate 2 classes perfectly while ensuring a good performance of the model. The size of the margin is tuned by a parameter C, which is proportional to the number of observations that can violate the margin. If C is large, the margin will be large (Figure 9, left), if it small, the margin will be narrow (Figure 9, right). In the latter case, the hyperplane will more fit to the training data.

SVC can also fit a non-linear decision function by enlarging the feature space of p dimensions using combinations of the features. Then, SVC fits a linear hyperplane in the enlarged feature space, resulting in a non-linear separator in the initial feature space.

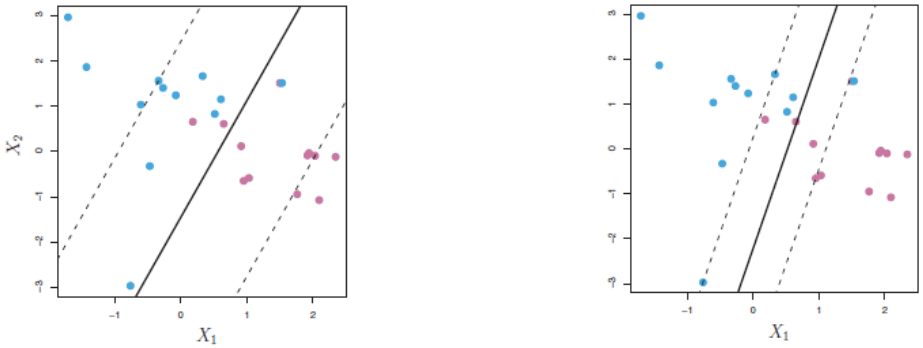


Figure 9. Support vector classifier. The hyperplane corresponds to the solid line and the margins corresponds to the dashed lines. The dots of the same color correspond to the observations for one class. $\boldsymbol{X_1, X_2}$ correspond to two features. It is a 2D feature space for which each observation is assigned 2 values, one for each feature. (James *et al*, 2000)

Support vector machine (SVM)

SVM differs from SVC by the way the feature space is enlarged, SVM does not use combinations of features but a function called kernel. By using Lagrangian multipliers to determine the coefficients for the p parameters, it appears that only knowledge of the inner product between all the pairs of observations is required (Eq. 3). x and x' correspond to 2 observations (pair).

$$\langle \mathrm{x}, \mathrm{x}' \rangle = \sum_{k=1}^{p} x_k {x'}_k \tag{3}$$

The equation of the linear support vector machine hyperplane is given in Eq. 4. $\alpha_i, \ldots, \alpha_n$ correspond to the coefficients for the n observations, β_0 to the intercept, and x to the observation to classify. In a binary classification problem, the label of the positive class is set to +1 and the label of the other (negative) class to -1. An observation x that is presented to the classifier will be assigned to the positive class if the function returns a positive value.

$$f(x) = \beta_0 + \sum_{i}^{n} \alpha_i \langle x, \mathrm{x}' \rangle \tag{4}$$

Knowing this, we can move beyond linearity by using a generalization of the inner product which is called a kernel function. A kernel function noted $K(a, b)$ is a measure of the similarity between two objects: a, b. The kernel can be either linear (Eq. 5) or non-linear (Eq. 6). $\varphi(x_i)$ corresponds to the observation x_i seen in the enlarged feature space φ.

$$K(x, x') = \langle x, x' \rangle \tag{5}$$

$$K(x, x_j) = \langle \varphi(x), \varphi(x') \rangle \tag{6}$$

In the case a linear kernel is used, a linear decision function will be fit, it will be non-linear if a non-linear kernel is used (Eq. 7). Examples of kernel functions are string kernels, which compute the similarity score of a pair of sequences. Four different string kernels are described in the Materials and Methods. Another non-linear kernel is the radial kernel, which requires tuning of another parameter besides C, γ. This parameter describes the distance up to which training examples have an influence for fitting the hyperplane.

$$f(x) = \beta_0 + \sum_{i}^{n} \alpha_i K(x, x') \tag{7}$$

The fact that only the inner product of the observations is needed allows to build a non-linear decision function that may derive from an infinite dimensional feature space φ. Moreover, the hyperplane is fit in a computationally efficient setting as we can operate in the enlarged feature space by only computing the kernel for each pair of observations.

On contrary to LR, SVM does not predict probabilities for the new observations and directly assigns the class. A method that can be used to compute the probabilities is to apply a sigmoid function that accounts for the distance between an observation and the hyperplane.

1.3.3. Stacking

In binary classification, stacking is a technique used to combine the predictions of two or more models (base models). The predictions of the base models are used to train another model that will give the final predictions. The predictions of the base model are referred to as meta-features. The advantage of this method is that it may result in better performance when each model taken separately performs better on a different subset of the data. Each model estimates the class to which an observation belongs. Then, the predictions are combined by a stacking model. This model will determine the best way to combine the predictions to result in a better (or equal) performance than each model would reach independently.

Stacking is also used to improve the performance in a multilabel classification problem by learning from the relations between targets. How this is done practically is explained in Materials and Methods. Those relations can be i.e. exclusions. In this example, the model used on top of the predictions of the first step may determine that the first label will be negative if the second label is positive. A negative relation exists between the first and the second label. Hence, the stacking model will assign a negative weight to the meta-feature (probability) of the second label when it must make predictions for the first label. In contrary, if a positive relation between the labels exist, the weight assigned to both predictions of the base models may be positive. Thus, if both labels are predicted to be positive by the base models, the certainty of the final prediction (after stacking) may be higher for each label.

1.3.4. Cross-validation

A simple method which can be used to evaluate the performance of a model initially separates the observation into a training set and a test set. The training set is used to train the model and the test set is used to evaluate how well the model performs on unseen data.

However, models often require hyperparameters to be tuned. Hyperparameters are user-defined values, they are not determined by the learning algorithm contrarily to the model coefficients (parameters). An example of hyperparameter is C of SVM. Several values must be evaluated in order to determine the one that may give the best performance. However, repeatedly training a model using a different value and assessing its performance on the test cannot be done. Indeed, determining the optimal value for a hyperparameter on the test data would consist in overfitting the model on this part of the data. In other words, the parameter is tuned such that it will perform well on this part of the data but not on other observations. Hence, this setup cannot be used to tune the parameters of a model. Note that overfitting also consists in including more parameters (features) than necessary in the model (Babyak, 2004). Including too many parameters will make the model more complex, it will follow the training observations too closely and will be less performant on new observations.

Another possibility is to randomly split the data into 3 datasets. One of them, the hold-out set, is kept apart from the data and the two others are used as explained above. One is used for training and the other is used to evaluate the performance for a given value of the hyperparameter. The set on which the performance is evaluated is called the validation set.

The value that gave the best performance on the validation set is then used to evaluate the model on the hold-out data. The problem here is that only part of the data is used as a validation set. This subset may not be representative of all the data, leading to high performance difference between the validation and the hold-out set. So, on what part of the data should the model be validated? In fact, it should be validated on all the available data, except the hold-out set, to have a reliable estimation of the best hyperparameter value. The method that allows us to do that is cross-validation.

The idea is to iteratively fit a model on the training set and evaluate it on the validation set until all the data has been used for validation. To do so, the dataset (except the hold-out set) is separated into k-folds i.e. k=3, 5, 10. At each iteration, a different fold is used as a validation set. At the end, the average performance across the k folds is computed. This is repeated for each value of the hyperparameter. The estimation of the average performance is more reliable than without cross-validation. Then, we can evaluate the performance on the hold-out set (test set) after training on the combined training and validation (tuning) set (Figure 10).

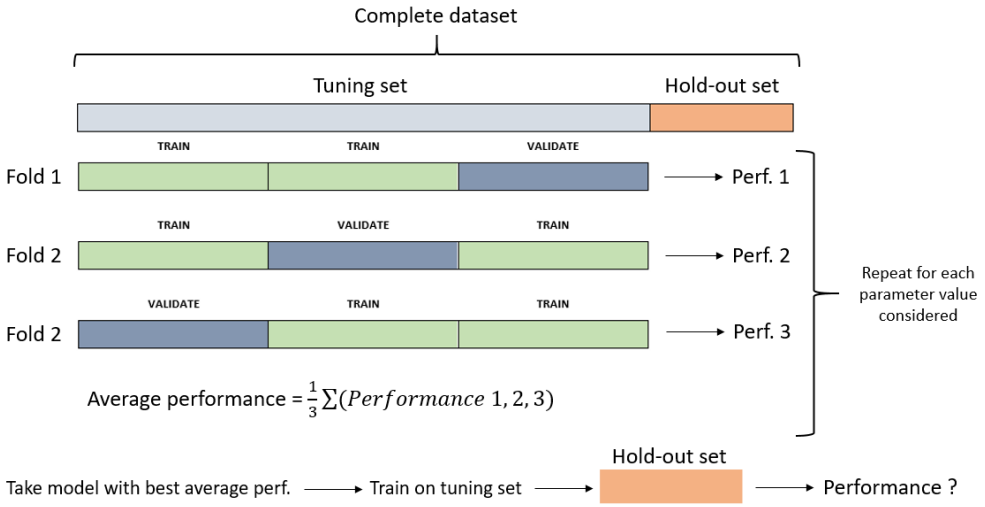


Figure 10. Cross-validation method for tuning parameters followed by the evaluation of the performance.

1.4. Transcription factor binding site identification

Transcriptional regulatory networks are directed graphs that represent regulatory interactions between transcription factors (TFs) and their target genes (Babu *et al*, 2004). Those networks are a straight-forward way to visualize transcriptional regulation. Moreover, they allow to analyze downstream or upstream effects due to a perturbation in a node. Building those networks first requires determining the specificity of promoters towards TFs. This can be done experimentally or computationally.

1.4.1 Experimental approaches

The experimental technique for the direct identification of TF binding sites (TFBS) is a combination of chromatin immunoprecipitation and sequencing (ChIP-seq) (Valouev *et al*, 2008). Chromatin is DNA complexed with proteins and RNA (Bernstein & Allis, 2005). ChIP-seq allows to study *in vivo* interactions between DNA and proteins. ChIP isolates portions of DNA

interacting with the proteins of interest (Nelson *et al*, 2006). The interactions between molecules in the cell (*in* vivo) are cross-linked so that transient interactions are permanently fixed. Afterwards, DNA is sheared so that regions interacting with proteins are not cleaved. Then, DNA regions interacting with the protein(s) of interest are isolated with specific antibodies. Cross-linking is reversed to purify DNA prior to sequencing. Sequencing allows to determine the sequences of the isolated DNA regions. In this case, the proteins of interest correspond to transcription factor interacting with DNA on TFBS. The sequencing step can be replaced by a DNA microarray (chip). However, this technique introduces bias and lacks reproducibility (Steger *et al*, 2011). Moreover, it requires to know the genome of the species under study beforehand. Sequencing can be performed *de novo* (Li *et al*, 2010).

Performing ChIP-seq/chip experiment is tedious and takes several days (Nelson *et al*, 2006). Alternatively, it is possible to predict the output of such experiments without the need of doing them. Computational methods allow to bypass wet lab work by making use of interaction data produced by previous experiments.

1.4.2. Computational approaches

Computational techniques for the identification of TFBS are divided into two main groups: *de novo* discovery of motifs and prior knowledge based identification. A motif is a pattern shared across binding sites. Discovery of motifs requires the knowledge of the sequences upstream the TSS of genes. Overrepresented patterns in promoter regions make it possible to determine motifs. Promoters sharing common motifs are likely to be co-regulated by the same TF. Based on that, a cluster analysis on motifs can be performed to give an overview of the regulatory network of the species under study (Ma *et al*, 2013). As this method does not require prior knowledge of interactions between TF and promoters, it is qualified as *de novo*. On the contrary, prior knowledge based identification of TFBS makes use of known interactions to train a model. Afterwards, the model is used to further scan new sequences for TFBS. ChIP can be combined with quantitative real-time polymerase chain reaction (qRT-PCR) to assess the binding of a protein of interest with predicted binding sites (Read, 2017). qPCR is more reliable as compared to a microarray experiment (chip).

Mutations appear at a certain rate in organisms. The mutations allow them to evolve and adapt to continuously changing environments. It is possible that certain organisms of a species become too different from their siblings and form two separate species. The genome of both sister species is different but still similar. Indeed, functional sequences, like genes or regulatory elements, evolve more slowly than non-functional ones (Cliften *et al*, 2003). Functional regions should thus be conserved across sister species. That is why, considering promoters, the functional regions, which correspond to TFBS, should be conserved (Down *et al*, 2007). Thus, besides construction of models, it is also possible to screen for TFBS by using a comparative genomics approach (Rodionov, 2007; Lenhard *et al*, 2003). Note that TFBS do not always correspond to regulatory elements. For example, a σ factor might bind a non-regulatory region if the sequence contains a binding sequence. Those TFBS should not be conserved across both sister species as they have no functional purpose.

1.4.3. Previous studies

There is a multitude of tools that have been developed to predict interactions (classification) or even affinity (regression) between σ factors and DNA sequences. Apart from the machine learning model that is used (or not), methods differ from each other by how sequences are transformed into features. Indeed, raw sequences cannot be used directly to build most models. Examples of features are: scores towards position weight matrices (PWM) (Foat *et al*, 2006), dinucleotide weight matrices (Siddharthan, 2010), k-mers (Annala *et al*, 2011) and pseudo k-mers (Lin *et al*, 2014). Dinucleotide weight matrices consider dinucleotides instead of one nucleotide. Some methods combine the different sort of features (Riley *et al*, 2015).

Position weight matrices (PWM)

PWMs allow to easily see and characterize the conserved motifs in a set of sequences (Xia, 2012). A PWM is a graphical representation of a set of sequences in matrix form: p rows and N columns, where N corresponds to the length of the sequences and p corresponds to the number of possible letters. Considering DNA, sequences are made up to four different nucleotides, making p equal to four. At each position of the PWM, the four letters are shown. The size of the letter allocated to a nucleotide depends on its frequency at this position across the set of sequences. Input sequences can be scored towards a certain PWM and classified as TFBS if the score is higher than a certain threshold. However, TFBS are not always conserved across different species and the promoters interacting with a given transcription factor do not always share common motifs (Scherf *et al*, 2000). The same importance is given to each input sequence as PWMs are frequency based. This method assumes that each nucleotide contributes independently to specific interactions. There is no consensus that determines whether this statement is true or not (Annala *et al*, 2011). Some studies showed independent contributions (Benos *et al*, 2002), whereas others showed interdependent contributions (Bulyk *et al*, 2002). Hence, PWMs are less reliable for predicting TFBS because of the issues explained above.

Figure 11. Example of the PWM of the promoters binding σ^{54}. (Cho *et al*, 2014)

K-mers approach

K-mer based methods circumvent the problems stated above. Indeed, they capture short-range interdependencies between nucleotides by taking substrings of the sequences into account (Wu & Bartel, 2017). A k-mer is a word of length k taken from the pool of all the possible words of length k. K-mer based methods create one feature per possible word (Table 2). As there are four nucleotides, 4^k features are created for a k-mer pool. Features are extracted from a sequence by looking at the occurrence of each word in this sequence. The dimensionality of the data gets high if large k-values are considered. Furthermore, the number of dimensions gets even higher if all the information about the order of the k-mers in the sequence is included in the model (Annala *et al*, 2011). However, excluding this positional

information allows not to make assumptions about position dependence, variable gap length between TFBS or multiple binding motifs (Weirauch *et al*, 2013).

Table 2. Example of 2-mer representation of sequences. One feature is created per possible 2-mer, resulting in a 16-dimensional feature space. The sequences are then represented by the number of occurrences of each 2-mer.

	AA	AC	AG	AT	…	CG	GA
AAGATAT	1	0	1	2		0	1
ACGATCG	0	1	0	1		2	1

Pseudo k-mers approach

Pseudo k-mers approaches do not lose the order information completely. The order of the k-mers in the sequence is approximated with a set of order correlated factors called θ (Chen *et al*, 2014b). Those factors are computed using physiochemical properties of the k-mers in a sequence and the proximity between k-mers. The first-tier correlation factor considers the most contiguous k-mers (shift of one nucleotide), the second-tier correlation factor considers the second most contiguous k-mers (shift of two nucleotides), etc. up to the $(L-K)^{\text{th}}$ -tier maximum (Figure 12). L is the length of the sequence and K is the length of the k-mer considered. The last tier considers maximum up to the $L-K$ most contiguous k-mers (shift of $L-K$). Thus, pseudo k-mers composition increases the dimensionality of the data by adding at most $(L-K)^* K$ parameters compared to a basic k-mers approach. Indeed, for each possible k-mer length, a maximum of $L-K$ parameters are added. A major limitation of this method is the availability of physiochemical parameters for the considered length of k-mers.

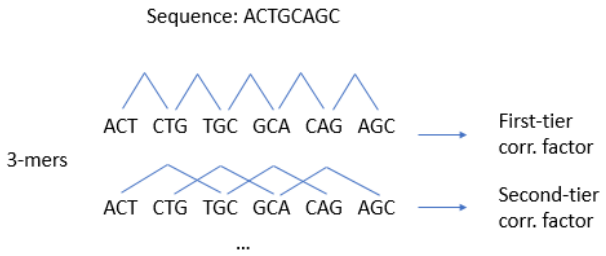


Figure 12. Example for the pseudo k-mers approach. The sequence is decomposed into the 3-mers that compose it. Then, the correlation (corr.) factors are computed based on the proximity of the sequences and physiochemical properties of the 3-mers. In this case $\boldsymbol{L} = 8$ and $\boldsymbol{K} = 3$. In this case, up to the fifth-tier correlation factor (8-3), which creates five more features for each sequence as compared to the k-mers approach.

BacPP tool

BacPP was developed in 2011. It is an example of a tool that can be used for σ factor specific assignment of input sequences (de Avila e Silva *et al*, 2011). The model was built with a dataset of 1034 promoter sequences from *Enterobacteriaceae* as positive instances. Those sequences had a length of 80 bp length. As negatives, a set of 1034 sequences of the same length were taken more than 80 bp upstream the TSS. They used this experimental setup to test the performance of their model. However, due to way the dataset was constructed, a bias may have been introduced, overestimating the model performance. Indeed, when evaluating the model for σ^x, they omit promoter sequences that do not interact with σ^x. Those sequences interact with other σ factors and resemble more to the σ^x-binding promoters than the

sequences in their negative set. Thus, the actual number of positives that are incorrectly predicted may be larger, which lowers the performance.
This tool makes use of a machine learning model called “neural networks”. This model was used to identify the ideal prototype for each σ factor. The prototype is represented by a PWM. Then, each nucleotide of a sequence is scored towards the prototype of a given σ factor. The scores are weighted by intervals with an integer (e.g. +2 for scores between 0.4 and 0.49, -1 for scores between 0.1 and 0.19). Lastly, a cutoff is used on the sequence score to discriminate between positive and negative observations. The cutoff is different for each σ factor. It is determined by looking at the intersection of the distributions of both classes. The distribution is plotted with the sequences’ score on the x-axis and the number of sequences with that score on the y-axis.

iPro54-PseKNC tool

iPro54-PseKNC was developed in 2014. It is a tool specialized in predictions of σ^{54} binding sites (Lin *et al*, 2014). It makes use of pseudo k-mers composition for vector characterization of the sequences together with support vector machines (SVMs). The physiochemical properties that are considered in this tool are related to the local structure of dinucleotides in the sequences; angular-twist, -tilt and -roll and translational-shift, slide and rise. These parameters have been calculated by Goñi *et al* (Goñi *et al*, 2007). A pseudo k-mers based approach produces a high number of parameters. Thus, there is a higher risk of overfitting. For this reason, a subset of the most discriminative features was selected. That is, features that individually allow to separate both classes the best. The positive set used consists of promoter sequences of 81 bp length going up to position +20 relative to the TSS. A negative set was created by taking sequences from intergenic and coding regions.

CHAPTER 2: AIMS

The purpose of this master thesis is to use machine learning to predict interactions between σ factors and DNA sequences. In this chapter, we present the different classification problems for which machine learning models were built. In the next pages of this work, we will first analyze the data and the labels to explain how the classification problems have been determined. Analyzing the data also allows to explain the performance of the different classifiers. Afterwards, we will present the performance of the models for each classification problem. In this chapter, we will already highlight how the problem was subdivided. An explanation of the two major classification schemes is given. Multiple models have been built to evaluate the performance of each type of classification problem separately. Training multiple models separately allows to determine key points for the problematic under study. The classification schemes were built to evaluate the performance of the models while combining their predictions for research applications. We will determine which classification scheme outperforms the other in the next chapter.

2.1. Promoters and non-promoters

When parsing DNA sequences, the first thing that can be accomplished consists in predicting whether a sequence is a promoter or not. A promoter will interact with at least one σ factor whereas non-promoter sequences do not interact with σ factors. This is a binary classification problem. A promoter sequence is defined as “positive” and is labeled with a 1, whereas a sequence that is not a promoter is defined as negative and is labeled with a 0.

2.2. Phase prediction

2.2.1. Activity of the sequence during the exponential phase

A sequence may be recognized by a σ factor during the exponential phase or not. It is also a binary classification problem. A sequence is active (positive) in the exponential phase if a σ factor interacts with it during the exponential phase. It is inactive if no σ factor interacts with it during that growth phase (negative). A model was built to classify DNA sequences based on their activity during the exponential phase.

2.2.2. Activity of the sequence during the stationary phase

A sequence that is not active during the exponential phase may be active during the stationary phase. Hence, a model was built to predict the activity during the stationary phase. In this case, a sequence that is active during the stationary phase is a sequence that interacts with a σ factor during the stationary phase. It is inactive if no σ factor interacts with it during that growth phase.

2.3. Interaction with σ factors

We have built three types of classifiers to assign σ factors to DNA sequences. That is, determining whether a σ factor interacts with a given sequence. One classifier predicts interactions during the exponential phase, another during the stationary phase and the last one predicts interactions with σ factors without regard to the growth phase.

2.3.1. Interaction while phases are grouped

For the prediction of interactions with σ factors while phases are grouped, the phase during which the interaction occurs is not considered. In this case, we only try to establish whether the recognition of the sequence by a given σ factor may occur or not. It is a multilabel classification problem because we can assign several σ factors (labels) to a given sequence. The labels are not mutually exclusive. A promoter may be recognized by all or few of the σ factors.

2.3.2. Interactions with σ factors in function of the growth phase

In most of the cases, a promoter will show a different interaction pattern with σ factors in function of the growth phase considered. Hence, we built a model that predicts interactions between a DNA sequence and σ factors for each growth phase.

2.4. Classification schemes to predict interactions with σ factors in function of the growth phase

The so-called classification schemes are in fact two models stacked on top of each other. Each model is trained to make predictions and, at the end, the predictions of each model are combined. The purpose for using the classifications schemes is twofold. Firstly, it allows us to analyze the effect that the predictions made in the first step have on the overall performance of the model. Secondly, it allows to improve the certainty of the predictions, and more particularly on the top predictions. The top predictions are the observations for which the model returns the highest probabilities. For instance, the second model may predict for a sequence an interaction with σ^{70} during the exponential phase. On the contrary, the first model may predict that the sequence is inactive during the exponential phase. On one side, it is classified as a positive but on the other side it is classified as a negative. Hence, the certainty on the final prediction for this sequence is low. Combining both models enables more robust predictions for further experimental analysis. It gives an idea about the usefulness of the model i.e. for research purposes. For instance, if one wants to screen sequences that could be used as synthetic promoters, it is important to determine what σ factors will interact with it and the growth phase during which the interaction will occur. It is possible that a model performs poorly on the overall data, but that top predictions are mostly correct. In this case, the researcher can extract the promoters from the top predictions and verify the interactions experimentally, on a smaller subset of sequences.

Two classification schemes are possible depending on what class is predicted first. The “phase-σ” classification scheme consists in determining the growth phase(s) during which a promoter is active first. Subsequently, σ factors are assigned to the promoters based on the growth phase(s) during which they are active (Figure 13, left). However, those predictions can be made the other way around. The “σ-phase” scheme starts determining which σ factors interact with a given DNA sequence. Then, for each σ factor interacting with this sequence, the growth phase during which the interaction occurs is predicted (Figure 13, right). For both schemes, we will evaluate and compare the performance after the combination of the predictions.

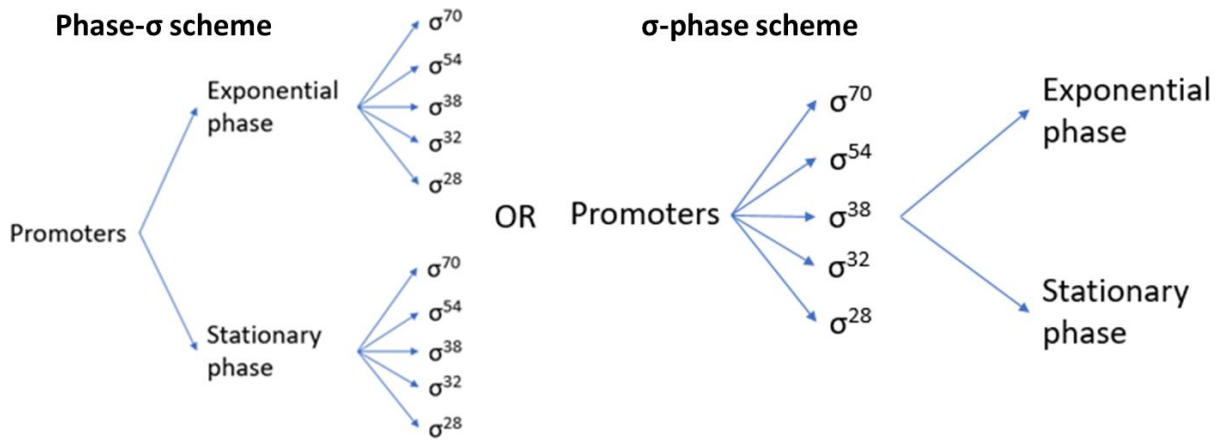


Figure 13. Overview on the classification schemes. Left: Phase-σ scheme. Right: σ-phase scheme.

CHAPTER 3: RESULTS AND DISCUSSION

3.1. Data analysis

In this section, we will describe the available dataset and analyze it. The analysis of the data was performed in order to determine the different classification problems. Moreover, it allowed us to find the reasons for the performance of each of the classification problems that were tackled. The dataset used is presented in the Materials and Methods.

3.1.1. Determination of the classification problems

In this subsection, we will analyze the data to consider the different possibilities for determining the interactions between DNA sequences and σ factors.

Promoters and non-promoters

The first model that was build is used to discriminate promoter from non-promoter sequences. The dataset of positives includes sequences that were not active for both growth phases considered. That is, they did not interact with any of the five σ factors during the exponential and stationary phases. Those promoters were removed from the dataset. The purpose of this classifier is to determine whether a 51 bp sequence is a promoter for at least one of the five σ factors studied (Figure 14).

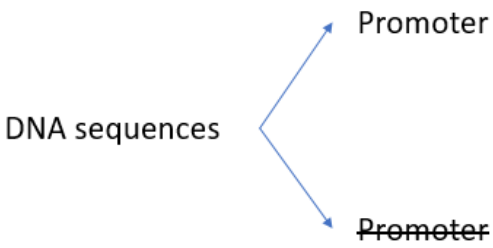


Figure 14. Promoter classification from sequences of 51 bp.

Overlap between σ factors

The data was also grouped by σ factor and "merged". That is, each promoter is said to interact with a σ factor if it does so during at least one phase. The number of overlapping promoters varies in function of the phase. Table 3 shows the number of overlapping promoters between pairs of σ factors for each growth phase and for grouped phases.

We can distinguish from two groups of σ factors Table 3. The first group represents the house-keeping σ factor and σ^{38}, which recognize most of the promoters binding to other σ factors (>90% and >60% respectively). The second group represents σ factors that recognize only few promoters interacting with other σ factors (<26%). This cluster includes σ^{54}, σ^{32} and σ^{28}.

Table 3. Percentage of overlap between promoters with regard to the σ factors by which they are recognized. a) Overlap during the exponential phase. b) Overlap during the stationary phase. c) Overlap when the exponential phase and the stationary phase are grouped. For instance, it can be read from table a) that 94% of the promoters recognized by σ^{38} are also recognized by σ^{70}. 60% of the promoters recognized by σ^{70} are also recognized by σ^{38}.

a)

Exp. phase		are also recognized by σ^{70}	σ^{38}	σ^{32}	σ^{54}	σ^{28}
% of promoters recognized by	σ^{70}	100%	60%	21%	9%	4%
	σ^{38}	94%	100%	24%	12%	5%
	σ^{32}	93%	67%	100%	8%	3%
	σ^{54}	90%	73%	18%	100%	7%
	σ^{28}	96%	85%	16%	19%	100%

Promoters recognized by		Total
σ^{70}	1808	1916
σ^{38}	1161	
σ^{32}	413	
σ^{54}	187	
σ^{28}	67	

b)

Stat. phase		are also recognized by σ^{70}	σ^{38}	σ^{32}	σ^{54}	σ^{28}
% of promoters recognized by	σ^{70}	100%	60%	22%	11%	3%
	σ^{38}	93%	100%	26%	13%	4%
	σ^{32}	92%	69%	100%	12%	2%
	σ^{54}	93%	72%	25%	100%	5%
	σ^{28}	93%	77%	14%	17%	100%

Promoters recognized by		Total
σ^{70}	2364	2517
σ^{38}	1520	
σ^{32}	560	
σ^{54}	279	
σ^{28}	81	

c)

Grouped phases		are also recognized by σ^{70}	σ^{38}	σ^{32}	σ^{54}	σ^{28}
% of promoters recognized by	σ^{70}	100%	60%	22%	10%	4%
	σ^{38}	94%	100%	25%	13%	5%
	σ^{32}	93%	69%	100%	11%	3%
	σ^{54}	92%	72%	22%	100%	6%
	σ^{28}	94%	80%	18%	20%	100%

Promoters recognized by		Total
σ^{70}	3299	3500
σ^{38}	2120	
σ^{32}	783	
σ^{54}	370	
σ^{28}	123	

The classification problem arising here is to build a model that assigns σ factors to DNA sequences correctly. That is, determining with which σ factor a given sequence interacts, for three different cases: the exponential phase, the stationary phase and "grouped" phases (Figure 15).

We can also see that contrarily to what has been explained in Subsection 1.2.4, the dominance of σ^{38} over σ^{70} is not increased when switching from the exponential phase to the stationary phase. Indeed, the ratio of the number of promoters interacting with σ^{38} over the number of promoters interacting with σ^{70} remains 64% during both growth phases.

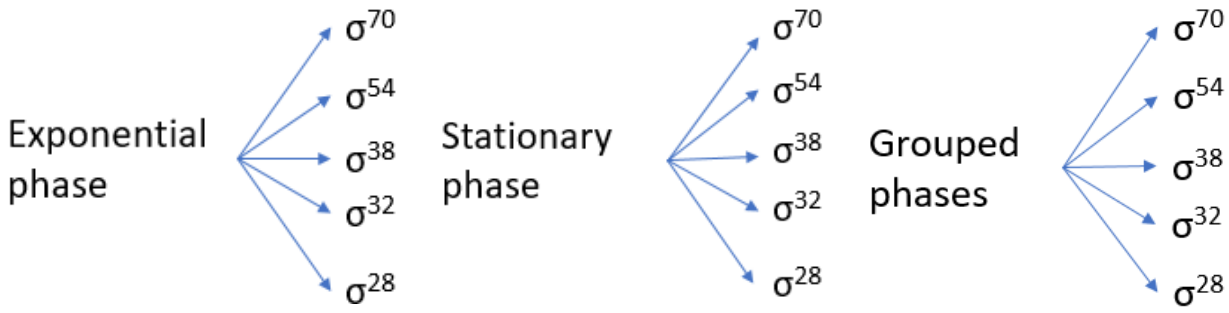


Figure 15. Assignment of the sequences to σ factors for 3 different conditions. We consider the exponential phase, the stationary phase and both phases grouped.

General overlap of the promoters between phases

Among the 3500 promoter sequences interacting with the σ factors under study, some are active during the exponential phase, others during the stationary phase and some during both phases. The Venn diagram below (Figure 16) shows the number of promoters that are active only during the exponential phase, the stationary phase or during both phases.

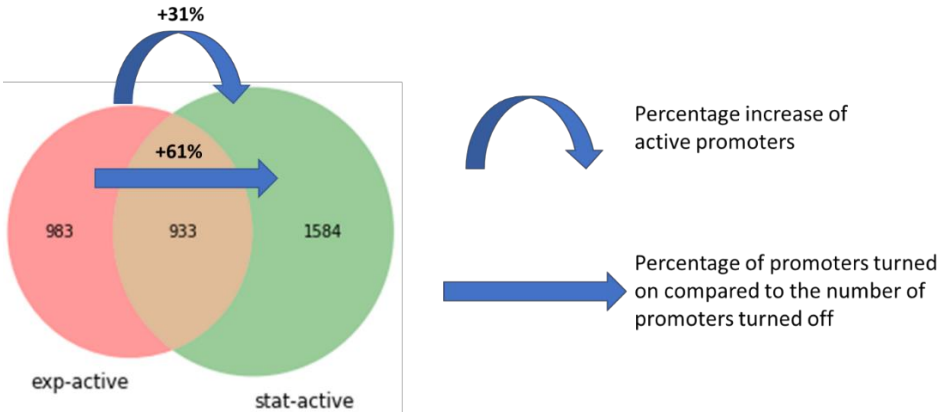


Figure 16. General overview of the activity overlap of promoters between exponential and stationary phase. The arrow at the top gives the increase in the number of active promoters while switching from the exponential phase to the stationary phase. The bottom arrow gives the proportion of promoters (%) that become active compared to the number of promoters that become inactive during the growth phase switch.

The number of promoters that are active during both growth phases is 933. More promoters are active during the stationary phase than during the exponential phase. Indeed, 1916 promoters interact with a σ factor during the exponential phase (983 + 933) and 2517 during the stationary phase. This represents an increase of 31%. The set of promoters that are active in each phase is different. The targeting of the σ factors towards a new set of promoters is called “promoter switching” and is visible in the data. Indeed, 983 promoters are abandoned by the σ factors while entering the stationary phase and 1584 new promoters become active. That is, 161% of the promoters are activated compared to the number of promoters inactivated. Thus, for 100 promoters abandoned while entering the stationary phase, 161 new promoters become active (+61%). As the set of active promoters change between growth phases, we will also build a model to classify promoters based on the phase during which they are active (Figure 17).

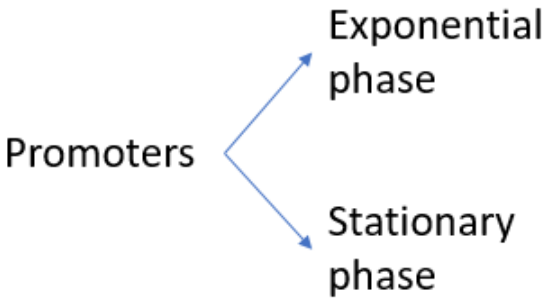


Figure 17. Assignment of promoters to the phase(s) during which they are active.

Detailed overlap of the promoters between phases for each σ factor

We will now look at the overlap of the promoters between phases into more details. We will analyze this overlap for each σ factor separately. Figure 18 gives a closer view on the transition between both phases with respect to each σ factor.

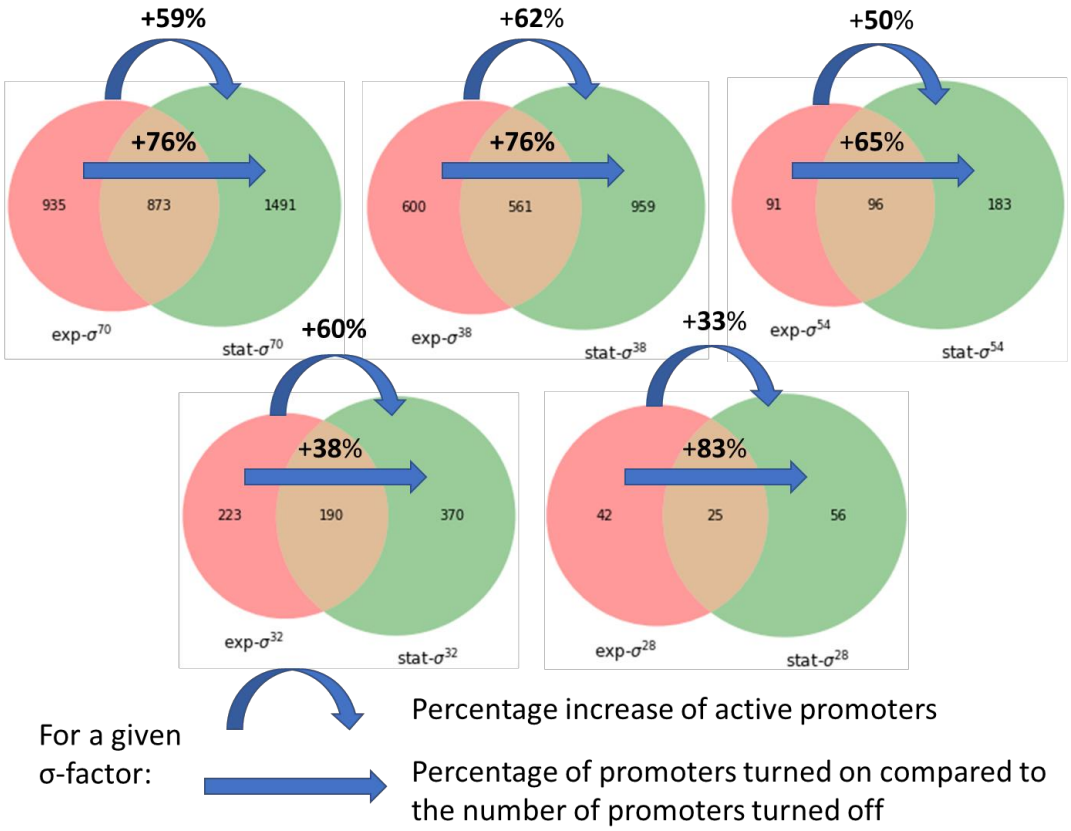


Figure 18. Detailed overview of the activity overlap of promoters for each σ factor. The red circles represent the number of promoters interacting with a given σ factor during the exponential phase. The green circles represent the number of promoters interacting with a given σ factor during the stationary phase. The overlap between the circles represent the number of promoters interacting with a given σ factor during both growth phases.

We can cluster σ factors into 3 groups regarding the quantitative and qualitative variation of the promoters that each σ factor binds while switching growth phase. The first group shows a high increase in the number of promoters it regulates (quantitative point of view) and high promoter switching (qualitative point of view). The second group shows a small increase in

the number of promoters it regulates but high promoter switching. Finally, the last group shows a small increase in the number of promoters it regulates and small promoter switching. The cutoff distinguishing high from small is at 50%. The first group includes σ^{70}, σ^{38} and σ^{54}. The second group includes σ^{32} and the third group includes σ^{28}. We have built a model to predict whether a σ factor interacts with a given sequence during a certain growth phase.

The dataset also shows that a promoter which is "deserted" by a σ factor while switching phase will not be recovered by any other σ factor. On the contrary, a promoter that becomes active during the stationary phase is usually recognized by at least another σ factor. This is the case for 42% of σ^{70}-binding promoters, 41% for σ^{38}, 34% for σ^{32}, 33% for σ^{54} and 31% for σ^{28}. The proportion of promoters binding several σ factors during the stationary phase is 66%. This proportion is the same as in the exponential phase. Next to that, the proportion of promoters that are active in both phases is only 27%. Thus, we believe that predicting first the phase during which a promoter is active and secondly the σ factors with which it interacts might result in better performance (Figure 19). Hence, two classification schemes were built to combine predictions from both steps (layers). For each layer, stacking was used to learn from relations between the labels. Those relations between σ factors are present in 66% of the cases during each growth phase. However, a relation should exist initially between the features that are used (sequences) and the classes (σ factors interacting with a given sequence). This will be analyzed in Subsection 3.1.2. The phase-σ scheme starts with the determination of the period of activity of each promoter. Afterwards, promoters are assigned to the σ factors with which they interact during this period. The σ-phase scheme starts with determination of the σ factors with which each promoter interacts. Subsequently, the growth phase during which the interaction occurs is determined for each σ factor.

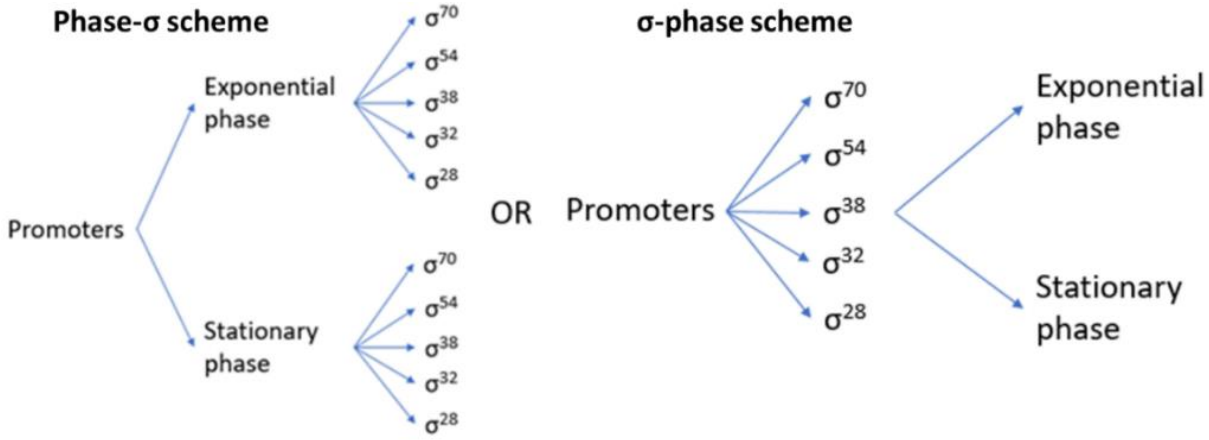


Figure 19. Classification scheme. Left: Determination of the period of activity of a promoter prior to the σ factor assignment. Right: Assignment of the σ factor of a promoter prior to determining when the recognition occurs.

3.1.2. Graphical analysis of the relation between sequences and classes

For each classification problem, the correlation of the sequences with their corresponding class was analyzed graphically with a dimensionality reduction tool (t-SNE). The practical explanations are given in the chapter Materials and Methods. The sequences could thus be shown in a 2D space. Thereafter, each observation was assigned to its corresponding label and

the separation of the different classes was observed with the crosses on those graphs. The crosses represent the barycenter of each class.

Relation between sequences and the class to be predicted

A graphical analysis allowed to analyze whether a correlation existed between sequences and their class. For the classes of σ factors, only promoters that bind a single σ factor were taken to allow specific labelling of each promoter. Those sequences are the more specific ones towards the σ factor with which they interact. If a relation between sequences and σ factors exists, the barycenters of each class should be separated in the plot (Figure 20).

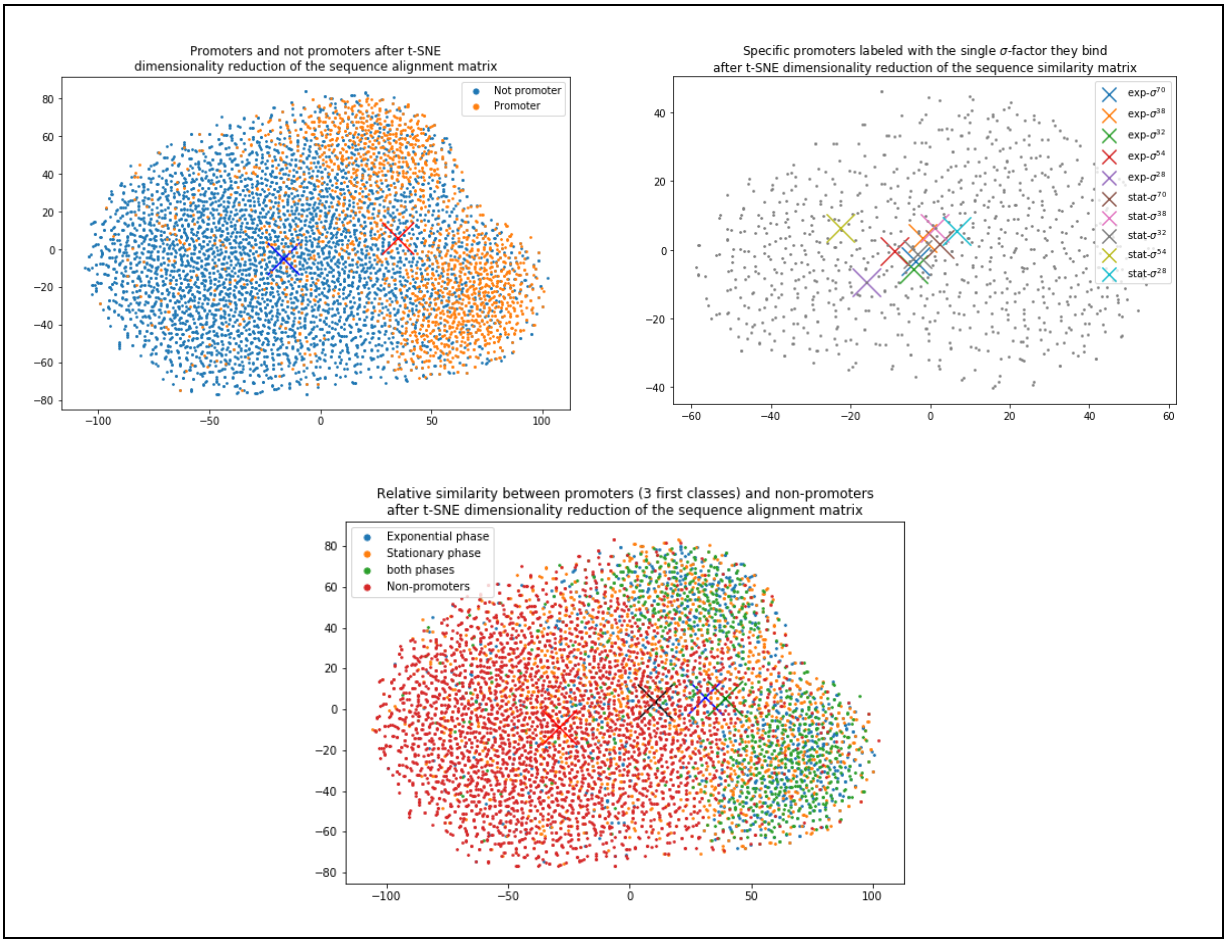


Figure 20. Graphical analysis of the relation between sequence and class. Left: both classes considered are promoter sequences and non-promoter sequences, right: classes are σ factors with the growth-phase specified, bottom: promoters are decomposed in 3 classes based on their period of activity. For clarity, the barycenter of the stationary phase-active promoters was marked with a dark cross.

Considering the promoter classification problem (Figure 20, left), we can see that each barycenter (one barycenter per class) seems to be separated from the other. This shows that the relative distances between promoters are smaller than the relative distances between promoter and non-promoter sequences. Indeed, t-SNE keeps observations that are close from one another in the initial space close from each other in the reduced space. This suggests that the sequences may be used to discriminate promoters from non-promoters. Indeed, the distribution of each class is not completely separated in 2D but may be separated in a higher dimensional space that we cannot visualize (4D, 5D, …). However, such plots give an insight into the ability of the string kernel to discriminate between classes.

Considering the plot for the classification of σ factors (Figure 20, right), the separation of the barycenters is less clear as compared to the first graph analyzed. However, a separation is still present. Also, the barycenter of a σ factor for a given phase is separated from the other phase. We believe there may be a link between a sequence and the phase during which a promoter interacts with a given σ factor. However, conclusions related to this link cannot be made here as the analysis does not include all promoters binding a given σ factor. This will be analyzed with the performance of the models.

Considering the prediction of the activity of a sequence in function of the phase (Figure 20, bottom), barycenters of each class seem to be well separated. The decomposition of promoters into 3 classes (stationary phase-active, exponential phase-active and active during both growth phases) shows that stationary phase-active promoters look more similar to non-promoters than the two other classes do. Hence, discriminating sequences that are stationary phase-active or inactive might be more difficult as compared to the same problem for the exponential phase. This hypothesis will be confirmed by the scores for the performance of the models that will be given in Table 5.

3.1.3. Graphical analysis for each classification scheme

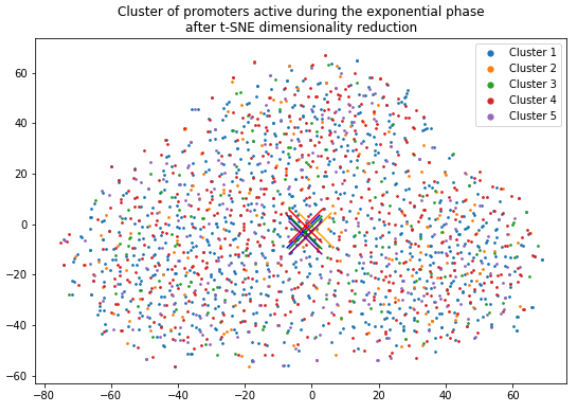
Here, we will consider both possibilities of the classification scheme depicted in Figure 19. The data that was used to make the plots for each classification scheme is explained in the chapter Materials and Methods.

Phase-σ scheme

We will first describe the "phase-σ" classification scheme (Figure 21). The first plot considers the promoters labeled on basis of the growth phase during which they are active. We can clearly see that the barycenters of each class are separated. Promoters that are active in the exponential phase (exp.-active) seem to differ from the promoters active in the stationary phase (stat.-active). Promoters that are active during both phases show more similarity towards exp.-active promoters than stat.-active promoters. Moreover, when looking at relative distances towards stat.-active promoters, exp.-active promoters seem to be closer than promoters active in both phases. An explanation for this might be that on one hand, both sets of single-phase active promoters show a certain specificity. But on the other hand, the specificity required for a promoter to be exclusively stationary phase-active is blurred when a promoter is active during both growth phases. Also, promoters that are active for both growth phases may resemble more to promoters that are exclusively exponential phase-active. Another explanation may be that the dataset has been incorrectly labeled. This hypothesis may be tested if other data is available.

Considering the two other plots, the clusters were formed based on the interaction pattern of the promoters with σ-factors for each growth phase (Materials and Methods). The barycenters of the clusters appear nearly on top of each other for the exponential phase-active promoters. Barycenters are more scattered for the stationary phase-active promoters. For the latter observation, this means that promoters can be grouped according to their σ factor interaction pattern. Those plots show that it might be more difficult for a model to assign σ factors to sequences for the exponential phase compared to the stationary phase.

This hypothesis will be refuted by the scores from the performance of the models that will be given in Table 8.



Sequence similarity of promoters:

- Top left: labeling based on period of activity.
- Top right: active during the exponential phase, labeling based on clustering results.
- Bottom right: active during the stationary phase, labeling based on clustering results.

The clusters obtained for each plot on the right-hand side are not the same. Indeed, the set of promoters on which clustering was performed is different. The plot at the top (bottom) is for exponential (stationary) phase active promoters.

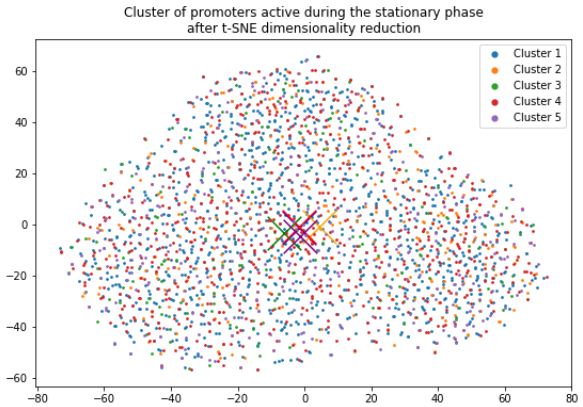


Figure 21. Graphical analysis for the "phase-σ" scheme.

σ-phase scheme

The first plot of the "σ-phase" scheme depicts barycenters on top of each other (Figure 22). Thus, sequences seem to be hardly grouped based on their interaction pattern. Note that here, the growth phase during which the interaction occurs is not considered when clustering. Indeed, the purpose of the first model in this classification scheme is to predict the σ factors interacting with a given promoter. The phase of growth during which the interaction occurs is considered at the next step of the scheme for each σ factor. The five other plots are used for the analysis of this next step. Those plots show that all σ factors except σ^{54} and σ^{28} have the 3 barycenters stacked on top of each other. Thus, for σ^{70}, σ^{38} and σ^{32}, no dissimilarity appears between the sequences interacting with them during different growth phases.

Contrary to the first classification scheme, the barycenter of the promoters active during both growth phases is as close as the other 2 barycenters (exp. active and stat. active) for each σ factor.

Comparison of the results of both classification schemes

From both graphical analyses, it can be highlighted that the dispersion of the barycenters seems to be higher when using the phase-σ scheme. This scheme might show better overall performance than the σ-phase scheme. Also, these results were produced using the equal elements string kernel and might have produced different results using another method for

feature creation. String kernels will be compared using the performance of the models that are built with each type of string kernel.

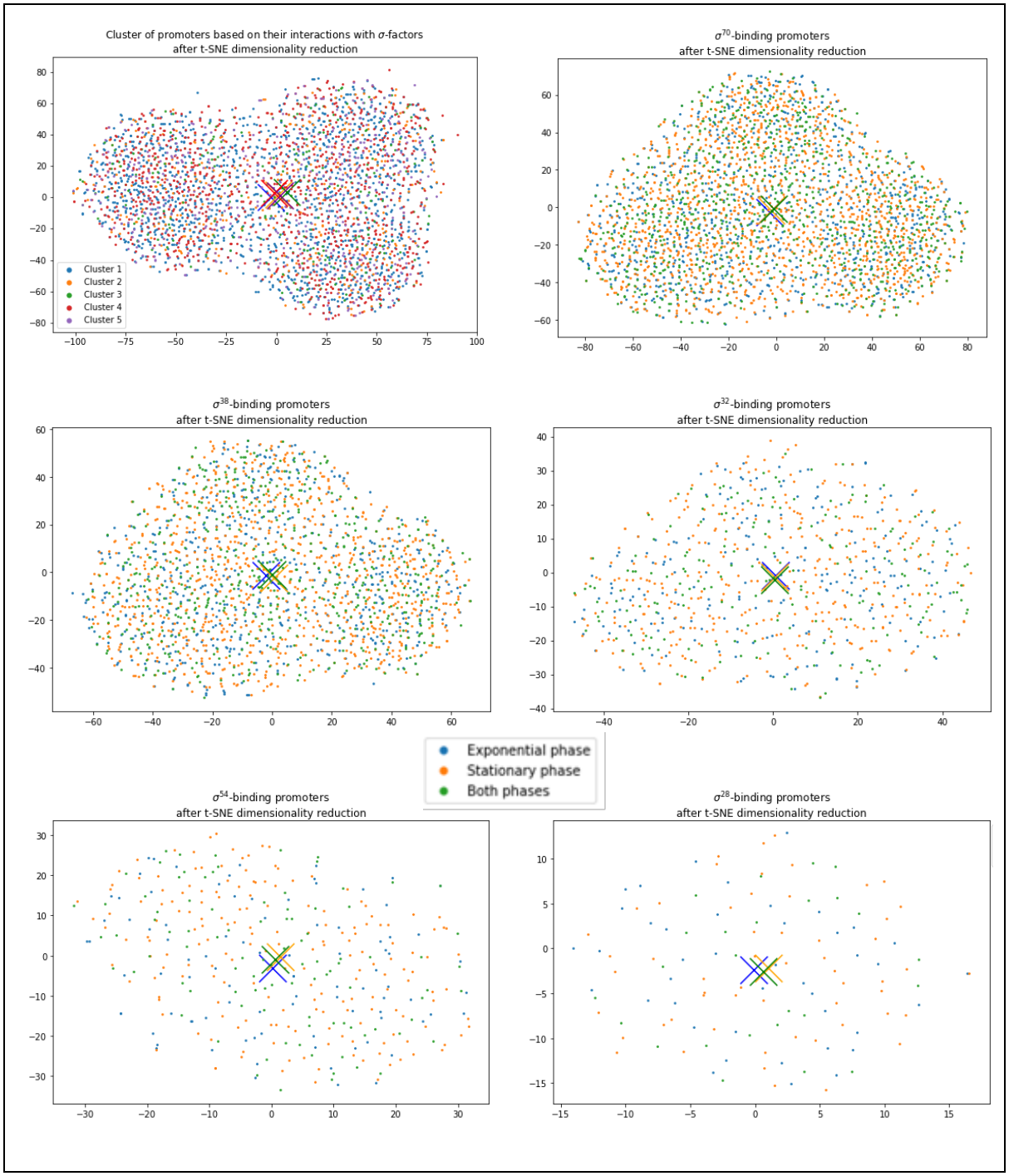


Figure 22. Graphical analysis of the "σ-phase" scheme. The graph at the top-left shows promoters labeled with the cluster to which they belong. The other graphs show promoters that interact with a certain σ factor. The promoters are labeled based on the growth phase during which they interact with this σ factor.

3.2. Performances for the different problems

In this section, we will present the test performances on the hold-out set of the different models (simple models) for each classification problem. Finally, we will compare the performance of both classification schemes, namely phase-σ and σ-phase. The performance expressed in terms of Area Under the Receiver Operating Characteristic curve (AUROC) is abbreviated by the term "AUC" in the presentation of the results. The explanations of the different performance measures, the extraction of the features from the sequences, the assignment of a threshold to the probabilities predicted and the experimental setups are given in Materials and Methods.

3.2.1. The models

For each classification problem, 11 models were trained. Three using logistic-regression (LR), and eight using support vector machine (SVM). One LR model was trained based on features extracted from the sequences (k-mers). The other 2 LR models were trained using the observations as seen in a 20D or 100D feature space after PCA dimensionality reduction (Materials and Methods). Considering the SVM models: three of them were trained using a linear kernel with the same setup as the LR models, one using a radial kernel, and four using two different string kernels: equal elements and weighted degree with shifts. Each string kernel exists of two different versions: the basic and improved version (Materials and Methods). There are two categories of models: k-mer based models and string kernel based models. For each problem, stacking was applied and the improvement of the performance was analyzed.

Presentation of the models' names

The name of the models with their abbreviation are given here. These abbreviations are used in the presentation of the results.

K-mer based models

- LR_U20: Logistic regression using the observations seen in a 20D feature space
- LR_U100: Logistic regression using the observations seen in a 100D feature space
- LR_BOW: Logistic regression using all the k-mers
- SVM_rbf_U20: Support vector machines using the radial kernel and the k-mers seen in a 20D feature space
- SVM_lin_U20: Support vector machines using the linear kernel and the observations seen in a 20D feature space
- SVM_lin_U100: Support vector machines using the linear kernel and the observations seen in a 100D feature space
- SVM_lin_BOW: Support vector machines using the linear kernel and all the k-mers

String kernel based models

- SVM_EqEl: Support vector machines using the equal elements string kernel
- SVM_EqEl*: Support vector machines using the improved version of the equal elements string kernel
- SVM_WDS: Support vector machines using the weighted degree kernel with shifts
- SWM_WDS*: Support vector machines using the improved version of the weighted degree kernel with shifts

3.2.2. Promoter prediction problem

Simple models

Here, we describe the performance of the simple models used to classify promoters and non-promoters. We can read from Table 4 that SVM_rbf_U20 is the least performant, with an AUC of 0.5 which is equivalent to a random decision. This model is followed by LR and SVM_lin which perform similarly, with an average AUC of 0.74 for LR and SVM_lin. The best performing models are the SVM using string kernels, with an average AUC of 0.84.

Table 4. Performance of the models for the promoter prediction problem.

Performance for promoter prediction											
	LR_U20	LR_U100	LR_BOW	SVM_rbf_U20	SVM_lin_U20	SVM_lin_U100	SVM_lin_BOW	SVM_EqEl	SVM_EqEl*	SVM_WDS	SVM_WDS*
AUC	0,72	0,74	0,77	0,5	0,72	0,74	0,77	0,85	0,85	0,83	0,82

Considering both LR and SVM_lin, including more features to the model slightly increases the performance. The gain in performance as compared to the number of features added to the model is especially higher when switching from 20 to 100 features (U_20 and U_100). Indeed, we can see from Figure 23 that the cumulation of the explained variance rapidly increases when considering up to the 100th PC (64%) and then it stabilizes.

The results for the SVM using different string kernels show slight differences (up to 0.03 AUC), with the advantage of EqEl on WDS. Also, both improved and standard versions of each string kernel show the same performances.

Figure 24 shows the variation of the average performance (AUC) on the tuning set when testing for different C parameters. We see that the choice of C barely influences the performance. An explanation for SVM is that the hyperplane that is fit in an enlarged feature space lies at the overlap between both classes and that the overlap is high. Hence, tuning C to allow little or more misclassification does not change significantly the shape and position of the hyperplane to result in a difference of performance.

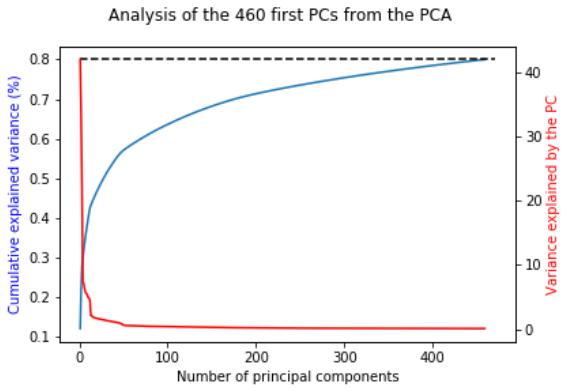


Figure 23. Analysis of the number of PC required to explain 80% of the variance in the observations. The x axis represents the number of first PC considered. The red curve represents the variance explained by each PC. The blue curve represents the cumulative variance (%) of the N first PC. For example, the cumulative variance when accounting for the 100 first PC is 63%. The 460 first PC are needed to explain 80% of the variance in the data.

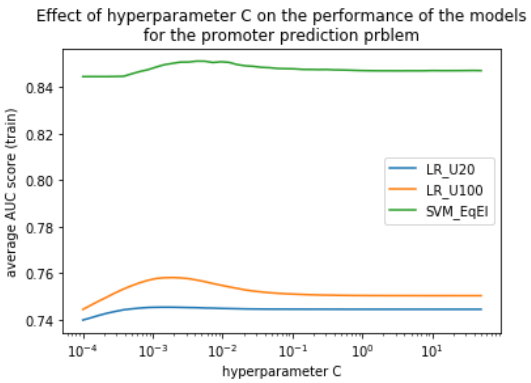


Figure 24. Effect of the C parameter on the performance of 3 base models on the tuning set. The y-axis gives the performance in terms of AUC. The x-axis represents C-parameter values on the range considered.

The poor performance of SVM_rbf is surprising as the graphical analysis from Subsection 3.1.2, (Figure 20, left) showed us that the distributions of each class do not overlap completely. Hence, an explanation of this result is that the choice of the default hyperparameter γ was inadequate. Indeed, after restarting computations for tuning γ, the AUC reached 0.72. Therefore, it can be informed that the choice of γ is of major importance for this kernel.

Stacked models

Stacking the predictions of LR_U100 and SVM_EqEI results in the same performance as with the best simple model (SVM_EqEI, AUC of 0.85). As there is no improvement in AUC score, we can deduce that SVM_EqEI simply outperforms LR_U100 on any subset of the data for this classification problem.

3.2.3. Phase prediction problem

We will now present the performances from the models used for the problem of determining the growth phase(s) during which a DNA sequence is active (multilabel classification). There are two labels, one describing exponential phase activity and a second one describing the stationary phase activity. Each label is binary as a sequence can be active (1) or inactive (0) for the growth phase (label) considered.

Simple models

Table 5. Performance of the models for the phase prediction problem.

PHASE PREDICTION											
AUC	LR_U20	LR_U100	LR_BOW	SVM_rbf_U20	SVM_lin_U20	SVM_lin_U100	SVM_lin_BOW	SVM_EqEl	SVM_EqEl*	SVM_WDS	SVM_WDS*
EXP	0,7	0,73	0,77	0,69	0,7	0,71	0,76	0,84	0,84	0,82	0,8
STAT	0,66	0,67	0,67	0,64	0,66	0,67	0,67	0,75	0,75	0,73	0,73

The results demonstrate that across all model types, the performance of determining whether a sequence is exponential active or not is better than for the stationary phase. The AUC scores are on average better by 0.08 for the exponential phase activity prediction as compared to the stationary phase. These results confirm the hypothesis that was made after the graphical analysis for this problem (Figure 20, bottom). Indeed, predictions for the stationary phase are more difficult to be made as compared to the exponential phase. This is because the similarity between stationary phase active promoters and non-promoters is higher as compared to the similarity between exponential phase active promoters and non-promoters.

As for the promoter prediction problem, string kernel methods perform better than k-mer based methods. The AUC difference is on average 0.07 for the best performing methods of each class. The remarks about the number of features included for the promoter prediction problem still hold. The difference in AUC when using the 20 first PC compared to all features is 0.05. Previous remarks are also valid for the comparison between string kernels: the EqEl string kernel performs slightly better than WDS string kernel (+0.03 AUC on average) and there is no difference in performance between the improved and basic version of EqEl string kernel. Nevertheless, there is a fractional difference between the basic and the improved version of WDS (+0.01 AUC).

A major limitation for the use of SVM_rbf and SVM_lin_BOW is the computational time. Considering the rbf kernel, each possible combination of the two hyperparameters (C and γ) must be tested to get decent results. SVM_lin_BOW models do not have this problem but the number of features included in the model is so big that computations become intractable, even if only the C hyperparameter must be tuned. For both the promoter prediction problem and the phase prediction problem, these two models did not outperform the best models (SVM with string kernels). Hence, because of limitations on time, we speculated that this would also be the case for the next classification problem and dropped those models.

Stacked models

Stacking the predictions of both LR_U100 and SVM_EqEl results in an insignificant increase in the performance for exponential phase activity prediction (+0.01 AUC). As discussed for the promoter prediction problem, we speculate that LR_U100 is outperformed by SVM_EqEl on each part of the data. A second explanation for this result is that stacking fails in finding

meaningful relations between the activities of promoters across growth phases. We did not expect this result as 27% of the promoters are common to both growth phases, which means that relations between promoters do exist. This may be due to the distribution of the different classes of promoters as explained for Figure 21, left.

Discussion on the performance of the stacked models

The results should be taken with care as non-promoter sequences are also included in the data. We believe that both types of the model can easily differentiate between promoters and non-promoter sequences. Indeed, the graphical analysis of this problem showed that non-promoters may be easier to discriminate from active promoters as compared to inactive promoters. Hence, including non-promoter sequences may bias the estimation of the performance as the $TNR = 1 - FPR$ is increased for a given TPR (true negative rate, false positive rate and true positive rate, Materials and Methods).

Next to that, we assigned a threshold to the probabilities predicted after stacking to analyze the type of predictions made by the model. The proportion of incorrect predictions is 26% when using the Hamming loss and 44% with the zero-one-loss. Moreover, the proportion of sequences classified as active for both growth phases represents 83% of the predicted active sequences. This is a lot as when accounting the fact that only 28% of the active sequences are active for both growth phases. Those results lead us to think that the model works well for sequences that are active for both growth phases but fails for single-phase active promoters. In other words, the model works well for determining if a sequence is a promoter or not but it cannot truly determine when the promoter is active.

We further pushed the analysis towards a multiclass classification approach. That is, only one label can be assigned of the 3 possible (exp. active, stat. active and active during both phases), which is the one for which the model has the most certainty. Hence, there is no need to set a threshold. As an example, an observation for which the model predicts 0.6 and 0.9 as probabilities to be active during the exponential and stationary phase respectively would be labeled as (1, 1) using a cutoff of 0.55 with the multilabel classification approach. But the multiclass classification approach may consider the label (0, 1) more suitable than (1, 1). This approach resulted in a Hamming loss of 21% (5% better) and zero-one-loss of 42% (2% better).

3.2.4. σ factor assignment problem

Now, we are going to present the performances of the models assigning σ factors to DNA sequences under the exponential phase, under the stationary phase and when growth phases are grouped. We will start with the description of the performances of the latter case (grouped phases) and cover the performances of each growth phase afterwards (exponential phase and stationary phase).

Grouped phases

Simple models

Table 6. Performance of the model for the σ factor assignment problem.

Performance for σ factor assignment									
AUC	LR_U20	LR_U100	LR_BOW	SVM_lin_U20	SVM_lin_U100	SVM_EqEl	SVM_EqEl*	SVM_WDS	SVM_WDS*
σ^{70}	0,74	0,76	0,71	0,74	0,76	0,83	0,83	0,81	0,81
σ^{38}	0,71	0,73	0,7	0,71	0,72	0,76	0,76	0,76	0,75
σ^{32}	0,66	0,65	0,66	0,66	0,67	0,69	0,69	0,69	0,68
σ^{54}	0,65	0,64	0,64	0,65	0,66	0,68	0,69	0,69	0,68
σ^{28}	0,64	0,58	0,66	0,63	0,62	0,65	0,65	0,69	0,67

The performance of the models used to assign σ factors to DNA sequences without accounting for the phase is given in Table 6. Overall, SVM_lin slightly outperforms LR but both types of model are outperformed by string kernel methods. The difference in performance between string kernel methods and k-mer based methods depends on the σ factor. When accounting for the best models of each category, the difference is 0.07 for σ^{70}, 0.03 for σ^{38}, 0.02 for σ^{32}, 0.03 for σ^{54} and 0.03 for σ^{28}. Considering U_20, LR and SVM_lin show similar performances. This is not the case for U_100, the AUC scores of SVM_lin are on average better by 0.03 as compared to LR when considering σ^{32}, σ^{54} and σ^{28}. The performances for the rest of the σ factors are the same. The performance for σ^{70} and σ^{38} drops when switching from LR_U100 to LR_BOW whereas it stabilizes for the other σ factors. An explanation for this may be that the model is too complex and starts overfitting on the tuning set for those σ factors. Indeed, we noticed that the average training performance raises by 0.01 AUC when including all the features whereas the test performance decreases by 0.05. These results motivate our hypothesis.

Across all the σ factors, the top scoring model is SVM_WDS. It performs similarly as compared to the EqEl type for all σ factors but the AUC score is better by 0.04 for σ^{28}. No difference in performance is recorded between EqEl and EqEl*. However, WDS outperforms WDS* by 0.01 AUC on average.

Stacked models

For this problem, LR_BOW and SVM_WDS predictions were stacked as those models showed the best average training performance across σ factors. As compared to the best simple models, the performance after stacking is increased by 0.01, 0.03, 0.01 and 0 for σ^{70}, σ^{38} and σ^{32} respectively. The performance does not change for σ^{54} and σ^{28} (Table 7).

Table 7. Performance of σ factor assignment after stacking (AUC score). The values on the left-hand side show the best performance across the simple models for a given σ factor. The values on the right-hand side show the performance after stacking.

σ^{70}	σ^{38}	σ^{32}	σ^{54}	σ^{28}
0,83 → 0,84	0,76 → 0,79	0,69 → 0,7	0,69 → 0,69	0,69 → 0,69

The ranking for the best-assigned σ factor is from first to last: σ^{70}, σ^{38}, σ^{32}, σ^{54} and σ^{28}. The difference in AUC across the models between the 1st and 2nd position, 2nd and 3rd, …, 4th and last is respectively: 0.05, 0.09, 0.01 and 0. The largest gap is found between σ^{38} and σ^{32}. We could make 2 groups of σ factors based on the performance for assigning them to a DNA sequence. The first group would contain σ^{70} and σ^{38} and the second one would contain the others. Those two groups coincide with the groups constructed from Table 3, which represents the overlap between promoters recognized by pairs of σ factors. Hence, we presume that highly overlapping σ factors are assigned more effectively as compared to σ factors with a low overlap with other σ factors.

Discussion on the performance of the stacked models

For this problem also, results should be taken with care as the dataset included non-promoter sequences. The performance for σ^{70} is very close to the performance of the promoter prediction problem. Indeed, 94% of the promoters interact with σ^{70}. Similarly, σ^{38}, σ^{32}, σ^{54} and σ^{28} interacts with 61%, 22%, 11% and 3% of the promoters respectively. Hence, we could conclude that the models are efficient for σ^{32}, σ^{54} and σ^{28} given the performance related to their assignment and the small proportion of the promoters they bind. Indeed, the performances show that even if a sequence is a promoter, those σ factors won't be "blindly" assigned to it whereas this can be done for σ^{70} and σ^{38}, in particular for σ^{70}. The task of assigning σ^{32}, σ^{54} and σ^{28} is more difficult as compared to σ^{70} and σ^{38} but the models manage to discriminate between promoters interacting with them or not, besides the non-promoters. We think that the proportion of inactive promoter should also be taken into account when evaluating the performance. Indeed, inactive promoters are more difficult to discriminate from active promoters as compared to non-promoters. This hypothesis will turn out to be true (except for σ^{28}). The proof will be given when comparing the performance of our models with the BacPP tool.

Exponential phase and stationary phase

Simple models

Table 8. Performance of the models for assigning σ factors during the exponential and stationary phases.

Performance for the assignment of σ factors during both growth phases										
	phase	LR_U20	LR_U100	LR_BOW	SVM_lin_U20	SVM_lin_U100	SVM_EqEl	SVM_EqEl*	SVM_WDS	SVM_WDS*
σ^{70}	Exp.	0,76	0,78	0,82	0,76	0,78	0,89	0,89	0,86	0,85
	Stat.	0,71	0,72	0,72	0,71	0,72	0,82	0,82	0,8	0,79
σ^{38}	Exp.	0,73	0,75	0,77	0,73	0,75	0,81	0,8	0,78	0,78
	Stat.	0,67	0,69	0,69	0,67	0,68	0,73	0,73	0,72	0,71
σ^{32}	Exp.	0,63	0,63	0,64	0,62	0,63	0,69	0,69	0,68	0,67
	Stat.	0,62	0,62	0,63	0,62	0,63	0,66	0,67	0,65	0,65
σ^{54}	Exp.	0,71	0,71	0,71	0,71	0,71	0,73	0,73	0,73	0,73
	Stat.	0,6	0,6	0,61	0,6	0,61	0,66	0,67	0,65	0,65
σ^{28}	Exp.	0,59	0,56	0,56	0,57	0,55	0,59	0,6	0,59	0,55
	Stat.	0,56	0,54	0,54	0,54	0,53	0,61	0,61	0,59	0,57

Table 8 presents the results for the assignment of σ factors to promoters for each growth phase. String kernel based models outperform k-mer based models. The models based on k-mers perform similarly (LR and SVM_lin) for a given number of features. This is the case for all σ factors except σ^{28}, for which LR outperforms SVM_lin by 0.02 AUC on average. Including the 100 first PCs instead of the 20 first PCs for a model (LR or SVM_lin) does not seem to affect the performance, except for σ^{28}. Indeed, increasing the number of features to 100 PCs decreases the AUC score for σ^{28} by 0.02 AUC on average for both SVM_lin and LR. However, the behavior of the LR models for each growth phase is different considering the variation of the performance while including all k-mers instead of the 100 first PCs. For the exponential phase, the performance for assigning σ^{70}, σ^{38} and σ^{32} increases by 0.04, 0.02 and 0.01 AUC respectively, but does not change for σ^{54} and σ^{28}. For the stationary phase, including all the features increases the performance of the models by 0.01 AUC only for σ^{32} and σ^{54}. This was also observed in the training results for both growth phases. Hence, we are not overfitting while increasing the complexity of the models. We saw that this was not the case for the assignment of σ factors when phases are grouped. We think that this is because the assignment of σ factors to promoters for a specific growth phase requires the model to be more complex as compared to when phases are grouped.

On average, EqEl string kernels outperform WDS by 0.02 AUC. There is no difference between the improved and standard version of EqEl but the standard version of WDS outperforms its improved version by 0.01 AUC on average. The remark for WDS is especially true for σ^{28} assignment, for which the performance of the standard version is better by 0.03 AUC.

For both growth phases, using the best SVM_string kernel instead of the best k-mer based model results in an increase of the performance by 0.05 AUC on average. The difference between both types of models is slightly more expressed for the stationary phase, for which the AUC scores are increased on average by 0.06. The difference of performance between the best models of each category is the most remarkable for σ^{70}, for which the AUC score differs by 0.07 and 0.1 for the exponential phase and the stationary phase respectively. For the other σ factors, the difference is of 0.04 AUC on average. These results lead us to think that string kernels combined with SVM allow to fit a better decision function (hyperplane) to separate the classes as compared to the k-mer based model.

The performances of the models for the exponential phase constantly exceed the scores for the stationary phase, except for σ^{28} when using string kernel based models. The largest differences are recorded for k-mer based models. Indeed, AUC scores are on average better by 0.06 and 0.04 for k-mer based models and string kernel based models respectively. The difference between the performances for each growth phase may be explained by the distribution of each class of promoters (Figure 25). The dataset related to each phase did not contain promoter sequences that are only active during the other phase. Hence, we see from this figure that removing the promoters that are only active for the phase that is not considered decreases the overlap between the positive and the negative class. The overlap with the distribution of the non-promoters is smaller for the exponential phase active promoters as compared to stationary phase active promoters. Thus, this overlap is decreased the most when removing the promoters that are only active during the stationary phase. therefore, the FPR should also be the most diminished. This may be an explanation for the difference between performances across growth phases.

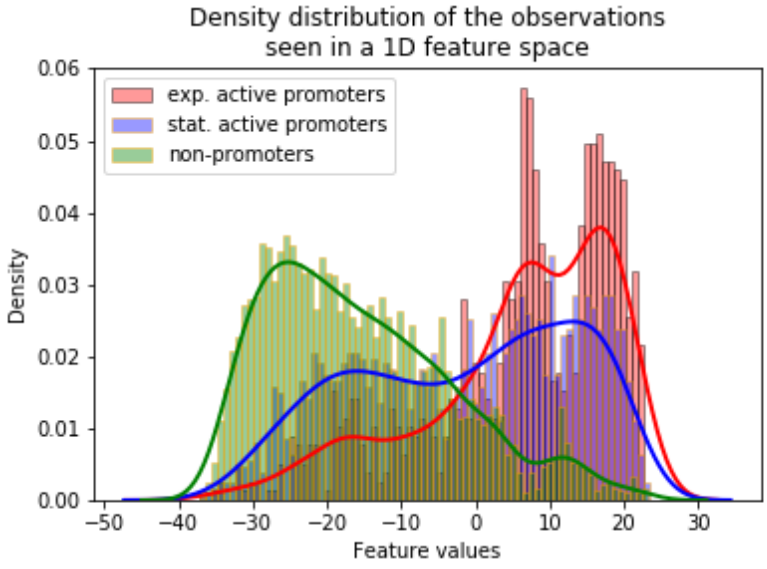


Figure 25. distribution of the sequences seen in a 1D feature space. t-SNE dimensionality reduction was applied on the similarity matrix resulting from the sequence alignments with the equal elements string kernel. The observations were reduced to a 1D feature space and the distribution of the values was plotted for each class. “exp. active” and “stat. active” labels represent promoters that are active during the exponential phase and active during the stationary phase respectively.

Stacked models

The best training scores were recorded for LR_BOW and SVM_EqEl for the exponential phase models and for LR_U100 and SVM_EqEl* for the stationary phase models. Therefore, those models were used for stacking. The results are given in Table 9.

Table 9. Performance of σ factor assignment after stacking (AUC score). The values on the left-hand side show the best performance across the simple models for a given σ factor. The values on the right-hand side show the performance after stacking.

	σ^{70}	σ^{38}	σ^{32}	σ^{54}	σ^{28}
Exp	0,89 → 0,91	0,81 → 0.82	0,69 → 0,71	0,73 → 0,76	0,60 → 0,63
Stat	0,82 → 0,82	0,73 → 0.75	0,67 → 0,68	0,67 → 0,70	0,61 → 0,62

The biggest improvement of performance after stacking is recorded for the exponential phase. The difference of AUC score improvement as compared to the stationary phase is 0.01. The improvement of the AUC score lies between 0.01 and 0.03 for the exponential phase and between 0.00 and 0.02 for the stationary phase. The effect of stacking is bigger for σ^{54} and σ^{28} as compared to the other σ factors for the exponential phase. Considering the stationary phase, stacking increases the AUC score the most for σ^{38} and σ^{54}. On the contrary, the performance does not change for σ^{70} and σ^{32}.

Figure 26 shows that the value for the parameter C has no influence on the average training performance for the tuning set. This was also the case for the promoter prediction problem and our hypothesis with regard to this problem is the same.

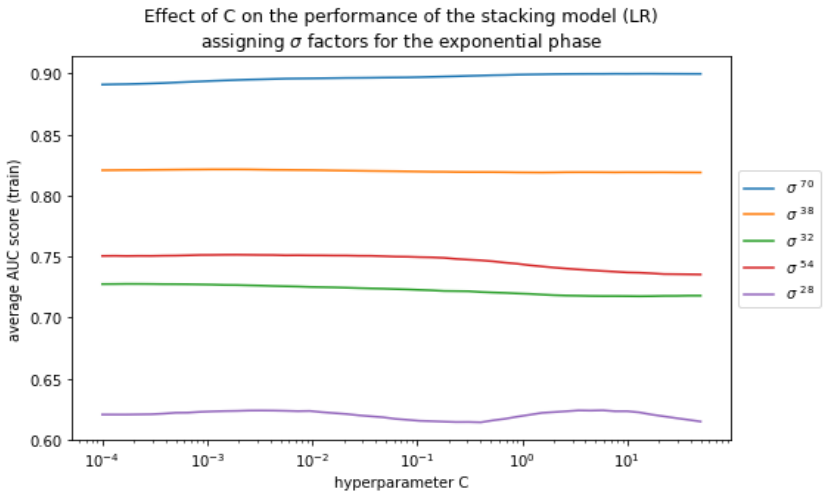


Figure 26. Effect of C on the performance of the model stacking predictions (LR).

The importance of the predictions of the simple models for assigning a σ factor to a sequence during a certain phase was analyzed. This allowed us to understand the behavior of the stacked model. The stacked model will consider the predictions made by the simple models for all the σ factors in order to improve the performance for a given σ factor. To this end, we analyzed the coefficients assigned to each prediction by the stacked model.

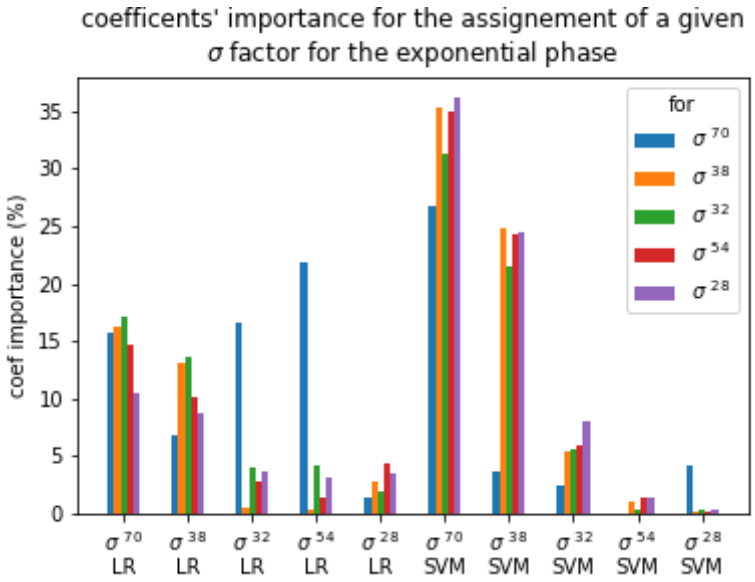
The coefficient assigned to a certain prediction can be either positive or negative. Therefore, we analyzed the absolute value of the coefficients to determine their relative importance. The importance of the coefficient for a σ factor reflects the relation found by the stacked model between that σ factor and the σ factor to be assigned. For instance, the coefficient that the stacked model assigns to the prediction of SVM_EqEl for the label σ^{70} may have 40% importance for predicting if a sequence interacts with σ^{38}. Figure 27 shows the result of this analysis for the predictions of the exponential phase and the stationary phase.

We can read from the bar plot for the exponential phase that, over all the predictions, the most weight is given to predictions for σ^{70}, σ^{38} and σ^{32}. This is the case for the predictions of both simple models. On average the importance assigned to the predictions of σ^{70}, σ^{38}, σ^{32}, σ^{54} and σ^{28} are 48%, 30%, 11%, 7% and 4% respectively. We noticed that the importance of the coefficients assigned to the predictions of σ^{32}, σ^{54} and σ^{28} for both models were the least important to predict an interaction between a sequence and one of those σ factors. However, this allows to increase the performance for their prediction by 0.03 AUC on average. The predictions resulting from SVM_EqEl on σ^{70} are considered to be the most important to predict the interaction between a sequence and any σ factor. An explanation for this is that most of the promoters recognized by a certain σ factor (>90%) will also interact with σ^{70}. Similarly, predictions made by SVM_EqEl on σ^{38} are the second most important for any σ factor except σ^{70}. σ^{38} is the second σ factor that recognizes most of the promoters recognized by other σ factors (>67% for σ^{32}, σ^{54} and σ^{28}). It is thus relevant that the stacked model mainly focuses on σ^{70} and σ^{38} to improve the predictions.

Overall, more importance is assigned to the predictions made by SVM_EqEl as compared to LR_BOW. The stacked model assigns on average 40% of importance to the predictions made on LR_BOW (60% for SVM_EqEl predictions). Hence, it is probable that the k-mer based model performs better on a subset of the data as compared to the string kernel based method. Otherwise, less importance would have been given to LR_BOW predictions as compared to SVM_EqEl. The stacked model values more LR_BOW predictions for the assignment of σ^{70} (60% importance) as compared to other σ factors. Also, the predictions of LR_BOW on σ^{32} and σ^{54} have a major importance to label σ^{70} (17% and 22% respectively, <5% for the other σ factors). The stacked model has probably found meaningful relations with regard to σ^{32} and σ^{54} as the performance for the assignment of σ^{70} is increased by 0.02 AUC. Moreover, the importance of the coefficients given by the model to the predictions of SVM_EqEl σ^{32} and σ^{54} are smaller than 3% for σ^{70} assignment. We do not understand this behavior as this is not observed for the stacked models related to the other σ factors.

Regarding the stacking model for the stationary phase, the relative importance of both simple models is more balanced as compared to the exponential phase. The importance given to LR_U100 and to SVM_EqEl* predictions are 46% and 54% respectively.

Exponential phase



Stationary phase phase

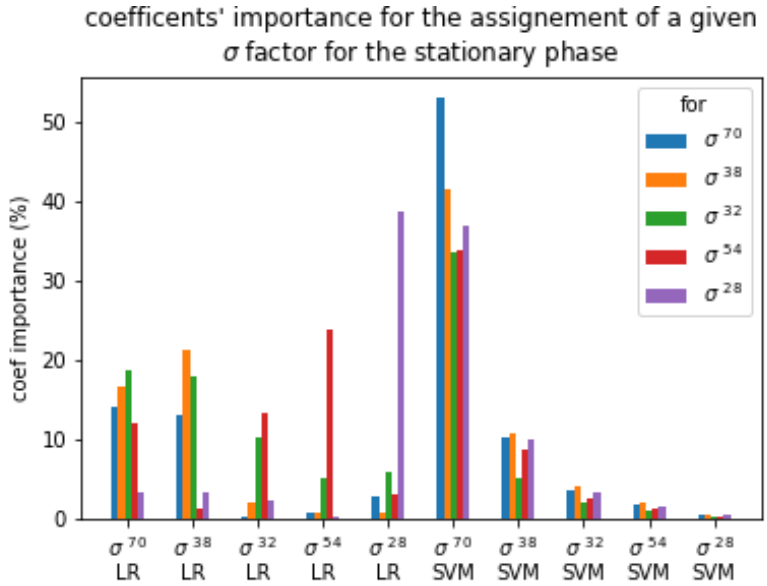


Figure 27. Importance of the predictions for each simple model (LR and SVM) for the exponential phase and the staionary phase. The bars represent the importance of the coefficient (in %). That is, the importance of the prediction made by the simple model for a given σ factor. Each σ factor has a specific color assignment. For a particular σ factor, the height of the bars of its color correspond to the importance of each coefficient. The x-axis represent the predictions considered. For example, we can read from the first bar plot that for the assignment of σ^{38} (orange bars) to a sequence, the predictions made by LR_BOW for σ^{70} and SVM_EqEl for σ^{38} have 16% and 25% importance respectively.

Comparison with BacPP tool

We will now compare the performance of our model with BacPP. In order to make their performance somewhat comparable to ours, we used the same experimental setup that the researchers used (4.1.3). First, we analyzed the "improvement" of the performances in terms of AUC to compare the scores with the performances that were obtained with the previous setup. The results are given in Table 10.

Table 10. Performances (AUC score) after stacking when removing promoters that do not interact with the σ factor considered. X → Y represent the performance before and after removing promoters that do not interact with the σ factor to be assigned.

	σ^{70}	σ^{38}	σ^{32}	σ^{54}	σ^{28}
Exp	0.91 → 0.91	0.82 → 0.91	0.71 → 0.87	0.76 → 0.89	0.63 → 0.76

On average, the performance is increased by 0.11 AUC. This confirms what we explained in the presentation of BacPP tool in the introduction. Also, it supports the hypothesis explained in the discussion of the performance for the assignment of σ^x in the "grouped phases" case after stacking. σ^x binding promoters are more difficult to discriminate from other promoters as compared to non-promoters. Hence, removing the promoters that do not interact with σ^x for evaluating the performance for σ^x assignment increases the performance. We think that this may be due to the fact that the overlap between the positive class (promoter) and the negative class (non-promoters or inactive promoters for a given σ factor) is reduced. According to those results, our thoughts considering the performance for σ^{32} and σ^{54} turned out to be true. Removing promoters that do not bind those σ factors results in performances such as that of σ^{70} and σ^{38}. However, this is not the case for σ^{28}. In conclusion, the problem of discriminating σ^x binding promoters from non-promoters show similar performances across all the σ factors in this setup, except for σ^{28}.

The comparison between both tools is given in Table 11.

Table 11. Comparison between BacPP performance and our performances. Acc. is the accuracy, Spe. is the specificity and Sens. is the sensitivity. Stack corresponds to our model.

	σ^{70}		σ^{38}		σ^{32}		σ^{54}		σ^{28}	
	Stack	**BacPP**	**Stack**	**BacPP**	**Stack**	**BacPP**	**Stack**	**BacPP**	**Stack**	**BacPP**
Acc.	0,848	0,805	0,848	0,866	0,772	0,923	0,815	0,952	0,656	0,971
Spe.	0,849	0,808	0,848	0,808	0,777	0,940	0,826	0,966	0,687	0,981
Sens.	0,847	0,803	0,848	0,924	0,767	0,907	0,804	0,938	0,625	0,962

Overall, BacPP works better than our model. This is especially true for σ^{32}, σ^{54} and σ^{28}. They display an accuracy, specificity and sensitivity that scores better by 0.2, 0.2 and 0.2 respectively as compared to our model for σ^{32}, σ^{54} and σ^{28}. However, this trend is reversed for σ^{70} and σ^{38}, for which our model scores better on accuracy and specificity by 0.01 and 0.04 respectively. However, the performance of BacPP in terms of sensitivity is higher by 0.02 as

compared to our model. The choice of a better threshold to determine each performance metric may have given different results. Those results are especially informative. No conclusions can be made as the dataset that was used is different.

We analyzed the graphical representation of the position weight matrices of the σ^x dependent promoters for each σ factor of our dataset to find an explanation for this difference (Appendix 1). It appeared that there is no true difference in the overrepresented motifs across the promoters specific for a certain σ factor. There are two explanations that may explain that. First, the promoter sequences might have not been properly aligned and it would explain the degeneracy of the PWMs. We think that this is not probable as the WDS string kernel would have shown better performance in that case. Indeed, WDS takes the shifts between aligned sequences into account. The second explanation is that the overlap between promoters recognized by any σ factor and σ^{70} is bigger than 90% for all the σ factors. σ^{70} specific promoters have in theory overrepresented motif around position -10 and -35. This means that the specificity of σ^x binding promoters may be blurred by the subset of the sequences that interact also with σ^{70}. However, this is unlikely given that it is not because σ^{70} binds a promoter recognized by another σ factor that this promoter contains the overrepresented motif(s). Hence, we believe that the most probable explanation would be that the assignment of σ factors to promoters after the ChIP-chip experiment has been unproperly performed.

3.2.5. General conclusions on the models

Number of features used

The complexity of the problem seems to have an influence on the performance of the models when describing the observations with more features. The assignment of σ factors to promoters for a specific growth phase requires more information in the data as compared to the same problem when phases are grouped. Similarly, the problem of classifying promoters and non-promoters is more complex when the phase must be accounted for. Hence, the performance is better when the sequences are represented by all the k-mers as compared to their 20D or 100D representation.

String kernel based models and k-mer based models

The models that use string kernels constantly outperform k-mer based models. K-mer based models perform overall similarly when using LR or SVM with the same data representation. The most simple kernel (EqEl) performs better than the more complex one (WDS). We speculate that this is due to the fact that the shift between a match when aligning sequences brings bias to the model as the sequences are aligned to the TSS. This may be a form of overfitting, as the information provided by this kernel is not biologically relevant anymore. A match that is shifted between two sequences (up to 5 positions) does not imply that those sequences behave similarly. That is, it does not mean that they will interact with the same σ factor or that they will be active during the same phase. The fact that the results are overall better using the greedy search version (WDS) of the kernel as compared to the exhaustive version (WDS*) (Materials and Methods) supports this hypothesis. Considering EqEl and EqEl*, we believe that the fractional difference in performance advantaging EqEl* may be due to the fact that there are no outstanding overrepresented motifs across the sequences of σ^x binding promoters.

Stacked models

Overall, stacking the predictions of the simple models does not seem to greatly improve the performance. We saw that this could be caused by the inability of the model to find meaningful relations between the labels. Also, because string kernel methods simply outperform k-mer based methods. It may be interesting to combine string kernels with other approaches such as BacPP as this may give better results.

The dataset

The fact that no difference between the overrepresented motifs were found across σ^x binding promoters is a key result that may explain the performance of the models for all the problems considered, except the promoter prediction. Indeed, if the labels of the sequences are not correctly assigned from the beginning, it may be hard to build a model that would work. The PWMs we obtained are contrary to the information that can be found in the literature. However, we cannot conclude this before testing our models with a different dataset.

3.3. Analysis of the classification schemes

In this section we will analyze the performance resulting from the combination of the predictions of each step. The way predictions were combined is explained in the Materials and Methods. We will first give an overview on the performance for the single model (no combination of the predictions). The single model is the one that is presented for the σ factor assignment for both growth phases. The only difference stands in the dataset that was used with the model. Afterwards, we will present the AUC scores that result from the combined predictions for each classification scheme and compare them with the scores obtained for the single model. Then, we will analyze the precision of the top predictions for each classification scheme.

3.3.1. Single model

The performances of the assignment of σ factors for the right growth phase when including only promoter sequences is given in Table 12. The scores correspond to the performances using stacked models (SVM_EqEl and LR_U100). We see that the performance is greatly affected (-0.15 AUC score on average) when removing non-promoter sequences from the data. This is because the probability that a randomly chosen positive (promoter interacting with σ^x) ranks above a randomly chosen negative (promoter that do not interact with σ^x and non-promoters) is smaller when non-promoter sequences are removed. As seen in the previous classification problems, non-promoter sequences are the easiest ones to discriminate from other sequences by the model. The average performance variation related to the predictions for the exponential phase and the stationary phase are -0.14 and -0.15 respectively. The performance for σ^{28} assignment for both growth phases is unaffected by the removal of non-promoter sequences from the data. An explanation for this is that the overlap between the distribution of σ^{28} binding promoters and non-promoter sequences is inexistent in the enlarged feature space where SVM_EqEl fits the separating hyperplane. Hence, removing the non-promoter sequences does not affect the FPR. For this setup, the best averaged performance across both growth phases results from the assignment of σ^{70} and σ^{28} to promoters.

Similarly, the performance for predicting the activity of a promoter during the exponential phase and the stationary phase decreases to 0.72 and 0.58 AUC respectively.

Table 12. Performance of the models for the assignment of the growth phase during which each σ factor interacts with a promoter. The first row represents the performance in terms of AUC score. The second row represents the variation of the performance (AUC) when removing non-promoters from the data.

	σ^{70}		σ^{38}		σ^{32}		σ^{54}		σ^{28}	
	Exp	**Stat**	**Exp**	**Stat**	**Exp**	**Stat**	**Exp**	**Stat**	**Exp**	**Stat**
Performance	0,71	0,54	0,64	0,56	0,55	0,54	0,59	0,54	0,64	0,62
Variation	-0,2	-0,28	-0,18	-0,19	-0,16	-0,14	-0,17	-0,16	0,01	0

3.3.2. Comparison between both classification schemes

Before presenting the results for both classification schemes, it is important to understand how they work. Each scheme combines two layers of predictions and only one layer differs from both schemes. In fact, determining the phase during which a promoter interacts with a certain σ factor for all the σ factors or determining the σ factors interacting with a promoter for each phase is the same problem. There are 10 labels to be assigned as there are 5 σ factors per phase and 2 phases per σ factor. The layer they have in common is the simple model described in Subsection 3.3.1. For phase-σ scheme, the other layer contains the predictions for the phase during which a promoter is active (2 labels). For the σ-phase scheme, the other layer contains the predictions for the σ factors interacting with a certain promoter (5 labels). The layers that are not common to each scheme are referred to as first layer. The one they have in common is referred to as second layer. The way the predictions were combined is explained in the chapter Materials and Methods.

In order to compare both classification schemes and the single model, we will first analyze the general performance of each scheme. We will also compare the performance of the schemes with regard to the single model. Finally, we will analyze the results for the top N predictions on the test set.

General performance

Table 13 indicates the performances of each classification scheme together with the performance of the single model. On average, the single model performs similarly as the phase-σ scheme on the test set (average AUC score of 0.59). However, the performance for phase-σ with regard to the assignment of σ^{28} for the exponential phase is lower by 0.11 as compared to the single model. We speculate that this result might be due to a lower performance for determining the activity during the stationary phase for σ^{28} binding promoters. Hence, combining the predictions for those promoters with the ones from the single model decreases the performance. Nevertheless, the performances of the assignment of all the other σ factors for any phase are on average better for the phase-σ scheme as compared to the single model. We believe that this might be due to a better performance for assigning the phase during which a promoter is active as compared to making the same predictions for each σ factor separately. Therefore, combining both layers of predictions improve the certainty on the final predictions. Across both schemes, the performance of the assignment of the σ factors during the exponential phase is always better as compared to the stationary phase. As explained for the results from Table 8, this seems to depend on the

distribution of the different types of promoters (active during the exponential phase, during the stationary phase or during both phases).

Table 13. Performance of each classification scheme. The two last rows indicate the performances of each classification schemes and the first row indicates the performance of the model described in the previous Subsection (3.3.1).

	σ^{70}		σ^{38}		σ^{32}		σ^{54}		σ^{28}	
	Exp	**Stat**	**Exp**	**Stat**	**Exp**	**Stat**	**Exp**	**Stat**	**Exp**	**Stat**
Single model	0,71	0,54	0,64	0,56	0,55	0,54	0,59	0,54	0,64	0,62
Phase-σ	0,73	0,55	0,64	0,55	0,57	0,57	0,59	0,54	0,64	0,51
σ-phase	0,58	0,53	0,58	0,55	0,53	0,52	0,53	0,51	0,54	0,55

The results show that the performances of the σ-phase scheme are close from a random decision. Indeed, the performance for assigning any σ factor for any growth phase is always smaller or equal to 0.55, except for σ^{70} and σ^{38} during the exponential phase. At this stage of understanding, we believe that this is due to the poorer performance of the first layer of this scheme (which assigns σ factors without regard to the phase) as compared to the first layer of the phase-σ scheme (0.75 to 0.85 AUC for the phase prediction and 0.69 to 0.79 for the σ factor assignment). The fact that the predictions of the first layer for σ-phase are low as compared to the ones for phase-σ scheme may explain that the overall performance is decreased while combining predictions.

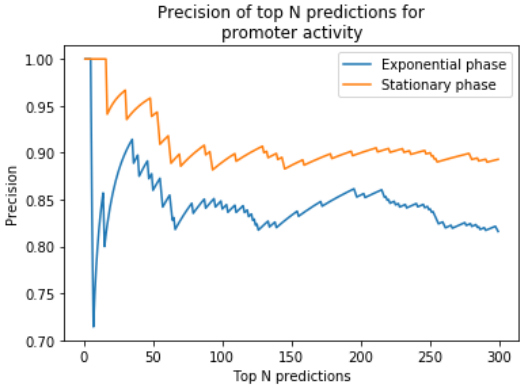
Precision of the top predictions for the first layer of each classification scheme

It is not because a model performs poorly on the average data that the performance on the predictions behave similarly. Indeed, the top predictions should be the ones for which there is the most certainty. Hence, even if there is a high overlap in the distributions of 2 classes, the observations that lie the farthest from the overlap may be classified correctly and with higher confidence. First, we analyze the precision of the top N predictions of the layer that both schemes do not have in common. That is, the phase prediction layer (phase-σ) and the σ factor prediction layer (σ-phase). The top predictions are positive predictions. They are a subset of N promoters for which the model has the most certainty that they are active for the phase considered or that interacts with a certain σ factor (phase prediction layer and σ factor prediction layer respectively). That is, the N promoters for which the predicted probabilities are the closest from 1.

We can see from Figure 28 that some labels are correctly predicted in the first layer whereas others are not. Considering the phase prediction layer (phase-σ), the top predictions for the activity during the stationary phase are 90 % correct up to the top 300 predictions. This is not the case for the exponential phase. The precision of the very top prediction is correct but 3 predictions out of 10 are incorrect afterwards. However, the next predictions are correctly predicted as positives with a precision around 83% until the top 300 predictions. The proportion of promoters active during the stationary phase and the proportion of promoters

active during the exponential phase in the test set are 54% and 71% respectively. This means that randomly selecting promoters from the test set as top predictions would have given 54% and 71% precision on average. Hence, both first layers perform better than a random decision. Selecting the 30 top predictions for each model results in 90% and 95% precision for the exponential phase and the stationary phase respectively. Thus, without regard to the σ factor with which a promoter interacts, a researcher can efficiently screen for 30 promoters that are active during the stationary phase and the exponential phase. Those results are good as this experimental setup is the one for which the estimation of the performance is the lowest. If non-promoter sequences would have been added to the data, the "random" model would have had a lower performance. Moreover, given the performance variation in terms of AUC shown in Subsection 3.3.1, our hypothesis is that the precision on the top predictions would not have been significantly affected.

Phase prediction layer (Phase-σ)



σ factor prediction layer (σ-phase)

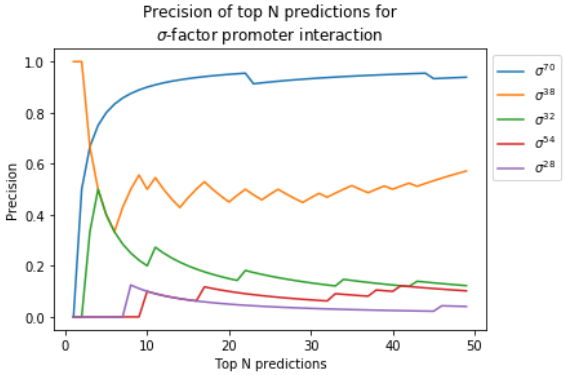


Figure 28. Analysis of the top predictions for the first layer of each classification scheme. The y-axis represents the proportion of correctly predicted interactions in function of the number of top predictions that are considered.

Considering the assignment of σ factors without regard to the phase, we see that it is correct in more than 50% of the top 50 predictions only for σ^{38} and σ^{70}. The very top prediction for σ^{70} is inaccurate but the next 20 ones are correct. The shape of this curve is only due to the fact that the prediction that ranked above all the others was a false positive. The precision is close to 95% for the top 50 predictions for σ^{70}. However, this result is not as good as it appears as the proportion of promoters that interact with σ^{70} is 94%. This means that randomly selecting 50 promoters from the test set and labeling them as positives would have been correct in 94% of the cases on average. Hence, we conclude that the model does not perform well for σ^{70}. This is the same for σ^{38}. Indeed, the precision on the top 50 predictions is close from 60% and the proportion of the promoters interacting with σ^{38} is 59%. However, the very top prediction is correct.

The performance of the top predictions for σ^{32} behave similarly as for σ^{70}. Taking the top 5 predictions for this σ factor results in a precision of 50%. As the proportion of promoters interacting with σ^{32} in the test set is 22%, we are of the opinion that the model may be used more effectively than a random model to screen for promoters that interact with this σ factor. However, further tests on another dataset should be made to confirm this hypothesis. The top predictions for the promoters that are said to interact with one of the other σ factors are also

inaccurate (σ^{54} and σ^{28}). Next to that, if the purpose of the researcher is to screen for sequences rather than promoters, we speculate that the precision of the model on the top predictions would not have been affected significantly as compared to the "random" model that would randomly pick top predictions from the data. In fact, we tend to believe that this model works well for determining whether a sequence is a promoter but not for assigning σ factors to promoters.

We speculate that those poor performances may be due to the fact that the distribution of the promoters interacting with one σ factor completely overlaps with the distribution of promoters binding to other σ factors. Moreover, each class may be evenly distributed inside the overlapping region. This causes the model to become unable to rank a promoter interacting with σ^x higher than a promoter that does not interact with it. Hence, there is no more certainty on σ^x binding promoters as compared to promoters that do not bind σ^x, making the top predictions unprecise. Figure 29 shows the overlap between the σ^x binding promoters and the density distribution of each type of promoters. We see that the distributions of the promoters binding to a certain σ factor greatly overlaps with promoters that interact with other σ factors. However, we cannot make conclusions on this plot which only considers 1D.

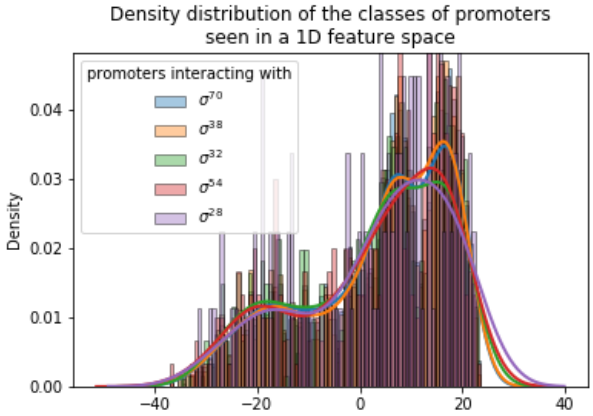


Figure 29. Overlap between the σ^x binding promoters. The x-axis represents the possible values in a 1D space after t-SNE dimensionality reduction.

Comparison of the top 30 predictions for each classification scheme and the single model

General precision

At his point, we are going to compare the top N predictions for each classification scheme and for the single model. That is, we are going to study the precision of the top N predictions resulting from the combination of the predictions of both layers. In this case, the 10 labels can either be positives or negatives. Hence, the precision was computed using the Hamming loss (Materials and Methods).

The results of the single model represent the performance when the predictions of the second layer (10 labels) are not combined with the predictions of the first layer. We are first going to compare the general performance of each type of model. The general performance is the precision of the top N predictions when all the labels are considered. Previously, we only considered the certainty towards one label. Hence, a top prediction is a promoter for which

the model has the most certainty about all the labels it was assigned. Afterwards, we will analyze the top predictions for each label separately (the 5 σ factors), for each growth phase.

Figure 30 shows the general performance of each type of model (single, phase-σ and σ-phase). The phase-σ scheme seems to perform better than σ-phase and the single model on the top 30 predictions. Indeed, ~65% of the labels are correctly assigned on the top 30 predictions whereas the precision is of ~55% and 45% for the single model and for the σ-phase scheme respectively. For the three cases, none of the top 30 predictions have all the labels correct, except for the 10th prediction of the σ-phase scheme.

The single model is the most balanced with regard to its predictions. For the top 30 promoters, positive labels are assigned exclusively to one growth phase in 50% of the cases. The σ-phase model assigns positive labels for both growth phases with a frequency of 90%. On the contrary, the phase-σ model is the most exclusive with regards to the interactions predicted in the top 30 predictions. Indeed, it assigns labels to only one growth phase in 100% of the cases. This does seem to depend on the combination of the predictions.

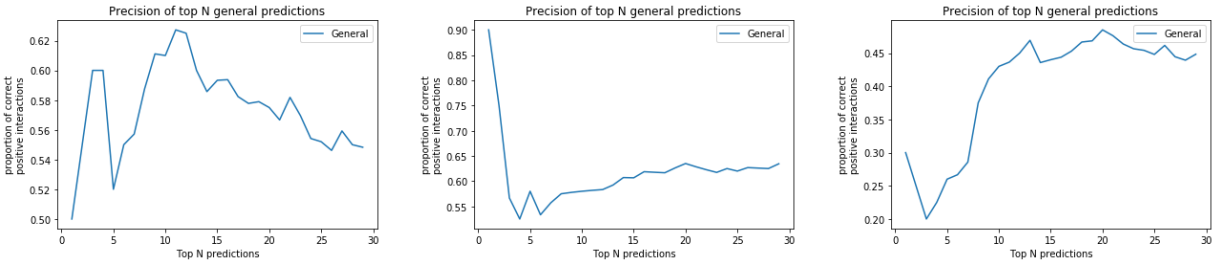


Figure 30. Analysis of the general precision on the top 30 predictions for each classification scheme and for the single model. Left: Single model. Center: Phase-σ scheme. Right: σ-phase scheme.

Precision for each σ factor on the top N predictions

Now we are going to analyze the precision of the top N predictions for each model across σ factors Figure 31. For each plot, N corresponds to the number of positive observations in the test set for the σ factor and the phase considered. Overall, the precision is better when using the classification schemes as compared to the single model. This proves that combining predictions allows to increase reliability of top predictions for each σ factor independently. Contrary to the classification schemes, the single model has a performance close to the random model or smaller than the random model in two out of the 10 cases. The difference in performance between the classification schemes and the single model is more important for the exponential phase. Considering σ^{32}, σ^{54} and σ^{28}, the top predictions are not precise (around 10%) for any of the models. However, there is always at least one model that performs better than random. Considering the top predictions for the exponential phase, the phase-σ scheme has a better averaged performance than the others. However, the σ-phase scheme performs better on σ^{38} for the stationary phase. Accounting for the very top predictions (~10%) during the stationary phase, the models that perform the best for σ^{70}, σ^{38} and σ^{32} are the phase-σ scheme, the σ-phase scheme and the single model respectively.

Overall, the precisions between σ factors are not comparable as the proportion of positive samples changes between σ factors. However, there is a greatest difference on the top 20

predictions between the random model and the classification schemes for σ^{70} and σ^{38} (around 20% precision difference).

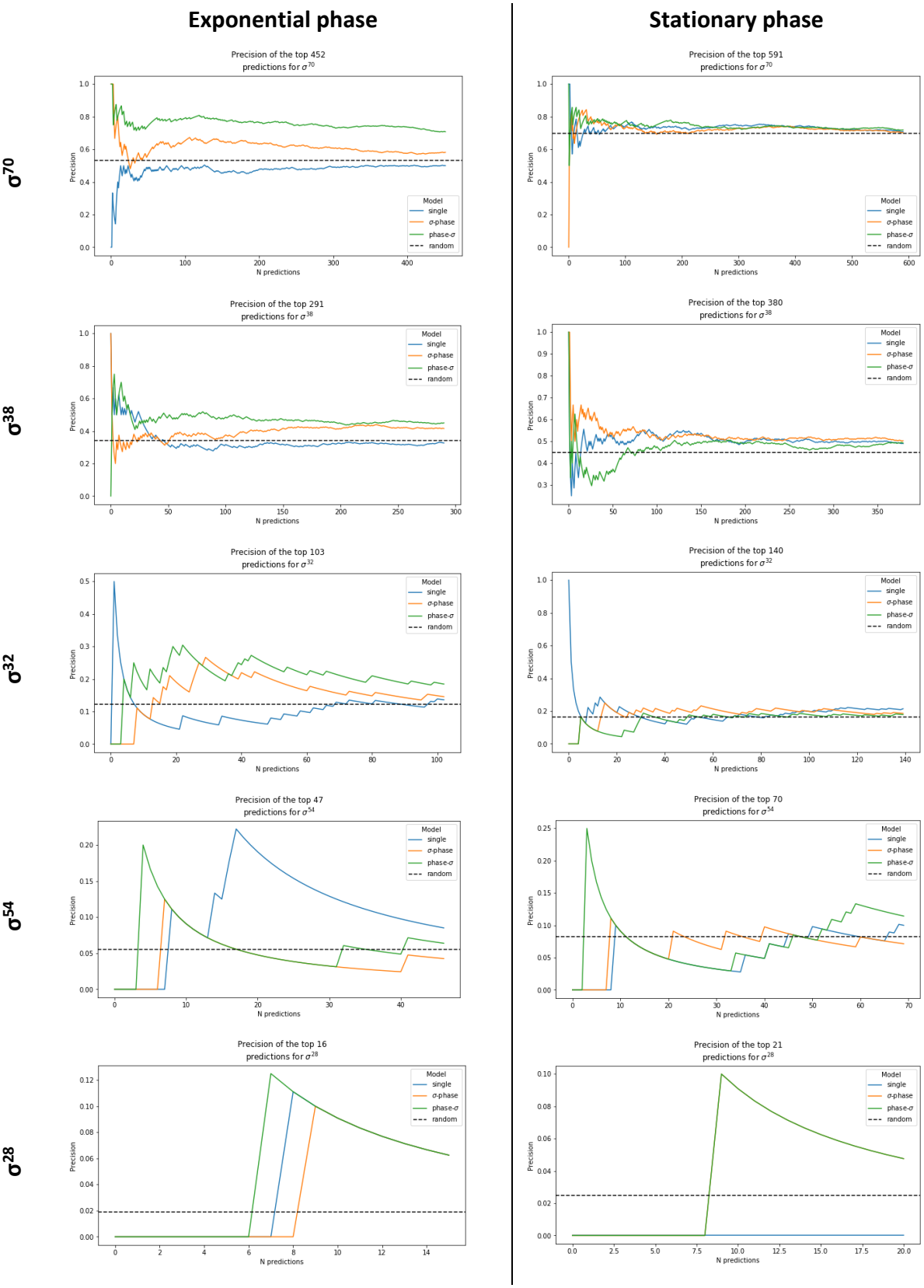


Figure 31. Comparison between the different models for σ factor assignment for both growth phases.

3.4. General conclusions

In conclusion, we believe that the models we built have proven their effectiveness for identifying promoters in *E. coli* based on a sequence of 51 bp. However, the problematic of assigning the right growth phase(s) during which a promoter may be active still needs to be solved. Also, the models used for the task of predicting the interactions between a promoter and all the σ factors did not prove great effectiveness given the results of the top predictions (zero-one-loss metric). We have seen that there was no significant difference between the overrepresented motifs of promoters binding to a certain σ factor. This is opposed to the information that is found in the scientific literature over the subject. Ideally, in order to achieve good performances, a dataset with differences between overrepresented motifs should be used. Accounting for that, we believe that our models may have allowed to accomplish both the assignment of σ factors to promoters and the prediction of the growth phase during which they are active. In this case, the predictions of the models could be used for the construction of a transcriptional regulatory network. However, this should be further confirmed using another dataset. Beside this, selecting a set of promoters that may interact with a given σ factor during a specific growth phase is made possible for σ^{70}, σ^{38} and σ^{32} by using the classification schemes. The latter increase reliability of top predictions as compared to the single model.

The models that use string kernels are more effective as compared to k-mer based methods. The models build on the latter method can be trained by using the observations transformed in a reduced dimensional space without significantly affecting the performance as compared to the computational efficiency.

The classification schemes increase reliability of top predictions as compared to the single model which performs better on the overall data. Hence, combining the predictions of the different layers is found to outperform the single model for research purposes, except for σ^{32} during the stationary phase. The phase-σ scheme is the best choice for screening promoters that interact with a certain σ factor during the exponential phase. Considering the stationary phase, screening for promoters that interact with σ^{70}, σ^{38} and σ^{32} should be performed using the phase-σ scheme, the σ-phase scheme, and the single model respectively. Any of the classification schemes can be used for screening promoters that interact with σ^{54} and σ^{28}. However, it would not be as effective as for the three other σ factors but it may already narrow the set of promoters that should be tested for the interaction experimentally.

For researchers who read this thesis and would like to apply our method, we propose the following pipeline to screen for promoters that may interact with a given σ factor during a specific phase. However, this should be done after training on another dataset.

1. Use SVM_EqEl to identify promoters in a set of sequences and extract all the predicted positives
2. Apply the classification scheme that match application on the predicted promoters
3. Take the top predictions for a given σ factor and test the interactions experimentally

CHAPTER 4: MATERIALS AND METHODS

4.1. Experimental setup

In this section, we present the data and the experimental setups that were used for each classification problem. In total, four different setups were used. We will describe each one of them and the reasons for which they were chosen. All the codes together with their output can be found on https://github.ugent.be/mmisonne/Thesis-Martin.

4.1.1. The dataset

The bacterial strain studied is *E. coli* K12 MG1655. The interaction between σ factors and their binding regions was determined by Cho *et al* (2014). They performed a ChIP-chip assay and processed the resulting data with a peak-finding algorithm to determine σ factor binding regions. Then, those regions were aligned to the TSS by using experimental TSS information. The final dataset with the information about the interactions between sequences and each σ factor was downloaded from the supplementary files of Cho *et al* (2014) (Additional file 7: Table S6).

The dataset consists of a positive and a negative set. The positive set contains 4724 promoter sequences of the *E. coli* strain whereas the negative set contains 50,000 non-promoter sequences. Non-promoter sequences derive from the *E. coli* genome. In the positive set, 3500 sequences interact with at least one of the five σ factors. The other 1224 sequences consist of promoters that were not active during the exponential or the stationary phase, or that were binding another σ factor than the five ones considered. Those sequences were removed from the dataset as they are of no interest with regard to the problems considered. The positive samples in the data consist of promoter sequences of 51 bp length aligned to the TSS. For each promoter, binary interaction information for five σ factors is provided: the house-keeping σ factor (σ^{70}) and four alternative σ factors (σ^{38}, σ^{32}, σ^{54} and σ^{28}). For each σ factor, interaction information is provided for the exponential phase and the stationary phase. This results in 10 labels (5 σ factors per growth phase) which are not mutually exclusive. Indeed, a promoter can be recognized by several σ factors and during both growth phases.

4.1.2. Classification of promoters and non-promoters, phase prediction and σ factor assignment

In this subsection, we describe the experimental setup that was used for evaluating the performance of the models for the classification of promoters and non-promoters, the prediction of the phase during which a sequence may be recognized by a σ factor and the prediction of the σ factors that may interact with a sequence. The same dataset was used to determine the performance of those classification problems. That is, we sampled 3500 negative sequences from the negative set to have an equal number of promoter and non-promoter sequences (7000 sequences in total). As a matter of fact, we want to evaluate the performance of the models when any sequence from the *E. coli* genome is presented to the models.

4.1.3. Assignment of σ factors for each growth phase

We have used two different datasets to evaluate the models for the prediction of interactions between σ factors and sequences depending on the growth phase. For the assignment of σ factors during the exponential phase, we sampled the promoters that bind to a σ factor during this growth phase and an equal number of sequences from the negative set (1916 sequences for each class). We did the same for the evaluation of the performance during the stationary phase (2517 sequences of each class). This was done to determine the performance of the model when screening sequences for determining if a sequence interacts with a given σ factor. The results showed that removing promoters that are not active for the growth phase considered may overestimate the performance. Hence, we used this more strict experimental setup for the final classification problem.

Comparison with BacPP tool

The setup we described in the previous paragraph is not the one for which the performance of the models is overestimated the most. We can also remove the promoters that do not interact with σ^x when assessing the performance for the assignment of σ^x to sequences. For instance, promoters that interact with any other σ factor except σ^{70} are discarded from the dataset when evaluating the performance of the model for σ^{70} assignment. The choice of the threshold for this problem is given in Subsection 4.2.2. We used promoters that were active for the exponential phase as the researchers evaluated the performance for this growth phase.

4.1.4. Evaluation of the classification schemes

The most strict approach for determining the performance of the models is to only use promoter sequences as those sequences are more similar to each other. We used this experimental set up for the evaluation of the performance of the classification schemes and the single model (σ factor assignment for each growth phase). Indeed, the purpose of this analysis is to screen a set of sequences and determine whether an interaction with a σ factor will occur and for which growth phase. Hence, non-promoter sequences were not included in the dataset in order to get a reliable estimation of the performance for this problem for research applications.

4.2. Performance evaluation

The way a model is evaluated depends on the purpose for which machine learning is used. In this section, we will present the performance metrics that were used and the reason for which a metric was chosen instead of another. Those metrics are used for classification problems. The output from a model is given in terms of probabilities of belonging to the positive class, as explained in the first chapter for LR and SVM.

4.2.1. Receiver Operating Characteristic curve (ROC)

True positive rate and false positive rate

In a binary classification problem, a model will predict positives and negatives. The true positive rate (TPR), also called sensitivity, refers to the proportion of positive observations that are recovered by the model (Eq. 8). That is, the proportion of correctly predicted positives, or true positives (TP) over the total number of positive observations (positives). The

number of positives can be calculated from the sum of TP and false negatives (FN). FN refers to negative observations incorrectly classified as such.

$$TPR = \frac{TP}{positives} = \frac{TP}{TP + FN} \tag{8}$$

The false positive rate (FPR), is the proportion of incorrectly predicted positives (FP) over the total number of negative observations (negatives) (Eq. 9). The number of negatives can be calculated by the sum of FP and true negatives (TN). TN refers to negative observations correctly classified as such.

$$FPR = \frac{FP}{negatives} = \frac{FP}{FP + TN} \tag{9}$$

ROC curve

In order to determine the performance of the model in terms of TP, FP, TN and FN, one needs to assign a threshold to the probabilities assigned by the model to the observations. Depending on the threshold that was chosen, the TP, FP, etc. will be different. As the FPR and TPR are interdependent, the choice of the threshold will depend on the purpose for which the model is used. A ROC curve allows to analyze the performance of a model without requiring a threshold. The curve is a plot of the TPR (y-axis) in function of the FPR (x-axis) resulting from the assignment of all possible thresholds. The area under this curve (AUC) is a direct estimation of the performance of the model. It gives the probability that a randomly chosen positive will rank above a randomly chosen negative. The AUC is not dependent of class imbalance, which makes this metric of particular interest in our case. Indeed, there are labels in our data for which there are much more positive observations as compared to the number of negative observations and vice versa. A perfect model will have a TPR of 1 for a FPR of 0 and results in an AUC of 1, there are no FN and no FP. Every instance in the data is correctly predicted. The ROC curve for such a model follows the y-axis while maintaining 0 on the x-axis. A model classifying observations randomly will have an AUC of 0.5. An example of ROC curve together with the AUC is given in Figure 32.

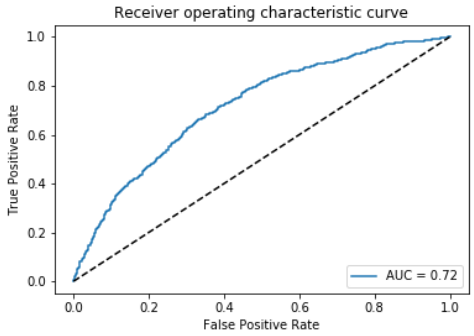


Figure 32. Example of ROC curve. LR_U20 on the promoter prediction problem.

4.2.2. Accuracy

The accuracy refers to the proportion of correct predictions, either positives or negatives. The comparison with the BacPP tool required to assign a threshold as the authors evaluated their model using the accuracy, the specificity and the sensitivity. There are plenty of methods for determining the optimal cutoff for a classifier and the choice is arbitrary. The method that can

be used to assign a threshold for the probabilities of a classifier consists in choosing the point where the TPR is high as compared to the FPR. That is, the threshold for which $|TPR - (1 - FPR)|$ is minimum.

4.2.3. Precision

Precision refers to the proportion of correctly predicted positive observations. The computation of this metric is different in a binary classification approach than in a multilabel classification approach. In binary classification the precision is given by Eq. 10.

$$Precision = \frac{TP}{TP + FP} \tag{10}$$

Multilabel classification case

In a multilabel classification approach the precision of the predictions can be computed with different metrics. We present here the zero-one-loss and the hamming loss.

The zero-one-loss gives the proportion of observations that do not have all their labels correctly predicted. It was used informatively to analyze the top N predictions of the single model and the classification schemes. This metric is too strict for this problem given that the performances of the models and that 10 labels need to be correctly assigned. A less strict metric is the hamming loss. It gives the proportion of incorrectly predicted labels for an observation. The precision over all the observations is given by $1 - Hamming\ loss$. That is, the proportion of correctly predicted labels. For instance, the precision for the prediction (0, **0**, 1, **1**, **1**) if the true labels are (1, **0**, 0, **1**, **1**) is 60%.

4.3. Feature extraction from the sequences

In this section, we present the methods that were used to extract features from the sequences as the latter cannot be used directly to train an SVM or a LR model. We used two methods: a k-mer based method and string kernels.

4.3.1. Extraction of k-mers from the sequences

For the k-mer based approach, we used a bag of word representation of the sequences in the training set. That is, we extracted all the possible subsequences of length 1 to 7. The maximal length was chosen after the literature study. We set the maximal length of the words to 7 in order to be sure to detect potential overrepresented motifs in the sequences. The bag of words representation creates one parameter per subsequence (word) found in the training set. Then, it creates a feature vector for each sequence which contains the occurrences for each word. This resulted in 21.735 features for the biggest dataset that was employed in this research (7000 observations). Hence, a dimensionality reduction approach was used to represent the observations in a smaller feature space. This allowed to increase the computational speed for fitting the models. Moreover, most of those features do not explain much variability and may thus not be useful to discriminate between different classes of observations.

Principal components Analysis (PCA)

Principal components analysis is an orthogonal transformation of the data allowing to represent the data in a reduced dimensional space of linearly uncorrelated variables. Moreover, those parameters point in the direction that explains the most variance between the observations. It was used to reduce the dimensionality of the problem and analyze the effect on the performance of the number of features used to build a classifier.

4.3.2. String kernels

A kernel is a function that describes the similarity between two observations. In this case, the kernel is a string kernel as it measures the similarity between aligned sequences. The result of the alignment of all the pairs of sequences in the observations with the string kernel is a similarity matrix of size $(N\ x\ N)$. This matrix gives the score for each pair of sequence. In this subsection, we present the four string kernels that were used for this master thesis.

Equal Elements (EqEl)

The EqEl is the most simple string kernel. It compares two sequences position by position and returns the number of occurrences of a match across all the positions. A match increases the score of an alignment by 1. Hence, the maximum score for an alignment is 51 (length of the sequence) and the lowest score is 0. As the sequences are aligned to the TSS, it should not be required to account for shifts between matches. A particularity of this method is that it does not take into account the position at which the match occurred. We acknowledge the fact that there are discussions as to whether positional information should be included to the model or not. Hence, we decided to build an 'improved" version of this string kernel that partially takes positional information into account.

Improved version of the EqEl (EqEl)*

In order to account for positional information, we screened the sequences from the training set and extracted the relative frequency of each nucleotide at each position. This is the same approach used as to build a PWM. A position that has low importance shows a similar frequency across all the nucleotides (0.25). On the contrary, a position that is important will show a higher frequency for one of the nucleotides (up to 1). Indeed, if a nucleotide is conserved among the promoters for a certain position, it is more likely that it plays a role for the interaction. Afterwards, the maximal frequency was extracted for each position and put into a weight vector. The only difference as compared to the standard version of EqEl is that each match (1) is multiplied by the weight relative to its position.

Weighted degree with shifts (WDS)

Improved version of WDS (WDS)*

The EqEl string kernel does not account for a potential shift in the aligned sequences. However, a sequence that is completely identical to the one with which it aligned but is shifted of one position will result in a very low similarity score. Therefore, we used a string kernel that was proposed by Ratsch *et al* (2005) called the WDS*. This string kernel takes both the length and the shift of a match into account to compute the similarity score (Figure 33). The maximal length of a match (word) and the shift allowed are both user-defined parameters. The WDS* string kernel is given by Eq. 11.

$$K(x_1, x_2) = \sum_{k=1}^{d} \beta_k \sum_{i=1}^{l-k+1} \sum_{s=0}^{S^{(s+i \leq d)}} \delta_s \, \mu_{k,i,s,x_1,x_2} \tag{11}$$

$$\mu_{k,i,s,x_1,x_2} = I(u_{k,i+s}(x_1) = u_{k,i}(x_2)) + I(u_{k,i}(x_1) = u_{k,i+s}(x_2))$$

Where x_1 and x_2 correspond to the 2 sequences aligned, d corresponds to the maximal length of a word, $\beta_k = 2(d-k+1)/(d(d+1))$ is the weight assigned to the match and depends of the length of the match, l corresponds to the length of the sequences aligned, S corresponds to maximal shift and $\delta_s = 1/(2(s+1))$ is the weight assigned to the match depending on the shift. μ_{k,i,s,x_1,x_2} indicates whether matches exist for a given position and shift when comparing x_1 with x_2 and x_2 with x_1. It can be equal to 0, 1 or 2 (no match, match only in one direction, match in both directions). $I(.)$ evaluates whether the equality in between brackets and returns 1 or 0 if it is true or false respectively.

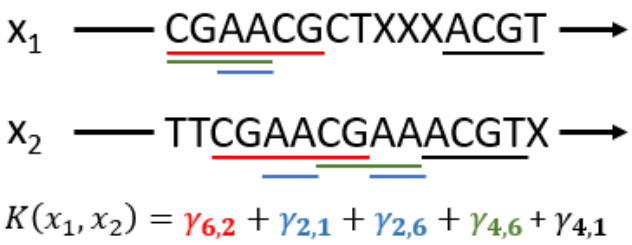


Figure 33. WDS string kernel illustration. $\gamma_{k,s}$ represents the contribution to the similarity score of the match of length k shifted from s between both sequences.

WDS* performs an exhaustive search of the matches for all the possible shifts and is therefore computationally intensive when considering larger shifts.

Basic version

The difference between the basic version that we propose and the actual WDS proposed by Ratsch *et al* (2005) is that only the first match is accounted for while parsing positions. In the example of Figure 33, a match of length 6 occurs between positions 1 and 3 of both sequences (shift of 2). Moreover, another match of length 4 occurs between positions 1 and 7 (shift of 6). This match is not accounted for in this more basic version of WDS*. It is a greedy algorithm for aligning 2 sequences faster as compared to WDS*. The limitation of this method is that a larger match which may be important for the classification problem can be missed.

For both WDS and WDS* a maximal word length of 7 was considered for the same reason as for EqEl. We arbitrarily chose a maximal shift of 5. Because multiple models and classification problems were evaluated, we decided not to investigate on the optimal choice for both parameters. This may be done in a further research.

4.3.3. Visualization of the data

We used the t-distributed stochastic neighbor embedding (t-SNE) to visualize the data in a 1D or 2D space (Van Der Maaten & Hinton, 2008). This tool allows to keep similar observations in the initial space close to each other in the reduced space. In contrast, PCA transforms the observation to keep dissimilar observations far from each other. Hence, we used this method

to have similar observations close from each other in the reduced space. This allows to get a more reliable estimation of the position of the barycenter of each class in 2D as the density distribution within each class of sequence is higher. t-SNE was applied on the similarity matrix built with the EqEl string kernel.

Graphical analysis of the classification schemes

For the graphical analysis of the classification schemes, the 3500 promoter sequences from the positive set were employed for the first layer of each classification scheme. Afterwards, the subset of promoters corresponding to the second classification step was taken.

Phase-σ

For the analysis of the phase-σ scheme, the first step (layer) consists in predicting the phase during which a promoter is active. Hence, promoters were labeled based on their period of activity. For the second layer, the σ factors with which a promoter interact during one phase are predicted. Hence, we took the subset of the promoters that are active during the exponential phase (resp. stationary phase) and labeled them with the cluster to which they belong. Indeed, one sequence can interact with several promoters. As we want one label per sequence, promoters were clustered based on their interaction pattern with σ factors and labeled accordingly. Hierarchical agglomerative clustering based on Ward's method and Euclidean distances was used and 5 clusters were formed. The principle behind this is that the interaction of a promoter with a σ factor is based on how close its sequence is from the "optimal" sequence. Thus, clustering promoters based on their interaction pattern makes sense. Indeed, sequences within a same cluster should have higher sequence similarity as they show similar interaction patterns.

σ-phase

For the analysis of the σ-phase scheme, the first step consists in predicting the σ factors interacting with a certain promoter. The second step consists in assigning the phase during which the interaction with a given σ factor occurs. Hence, promoters were labeled based on the cluster to which they belong for the same reason as for the second step of the phase-σ scheme. For the second step, the promoters that interact with a given σ factor where labeled based on the growth phase(s) during which they interact with this σ factor (exponential phase, stationary phase or both phases).

4.4. Selection of the optimal parameters

4.4.1. Parameter range

The optimal parameters (C, for both LR and SVM) were tested on the tuning set using a log scale going from [10^{-4}, 50] by taking 50 steps in between. The results showed that this was somehow exaggerated, taking 4 steps in between the same limits gives the same results. One limitation for testing 50 parameters for each model is the computational time. Moreover, the "L1" penalty and "L2" penalty were also tested when tuning LR models. They refer to different regularization methods to reduce the complexity of the model. L1 and L2 correspond to the L1-norm (not squared) and L2-norm (squared) loss functions.

4.4.2. Cross-validation with (multilabel) stratification

For all the problems, the training (tuning) and test set were split using a multilabel stratification approach to conserve the proportions of each class in both datasets. This allow to work with a training (tuning) set and test set that do not differ too much to have better estimation of the performance. Similarly, we used multilabel stratification to split the tuning set into k folds (stratified k-fold cross-validation, k=3) for minimizing the variance of the results in terms of performance across the k validation sets. Moreover, this technique makes it possible to obtain folds for which positives will always be present. Indeed, some of the labels have only few positives instances in the data (σ^{54}, σ^{28}). Using a random approach for splitting the dataset may lead to folds for which no single positive sequence is present. For each model, we extracted the parameter that gave the best average performance on the k validation sets of cross-validation. Afterwards, we trained the model with the optimal parameter value on the whole tuning set and evaluated it on the test set.

We are aware that nested cross-validation may have led to even more reliable results. However, we speculated that the size of our dataset combined to the stratification approach would allow reliable performance estimations to be generated. Hence, we decided not to investigate it.

4.5. Selection of the stacked models

For each classification problem, the predictions resulting from two models were stacked. We always used only one model from each type of models (k-mer based or string kernel based). Moreover, the models were chosen based on a combination of two properties. First, it had to outperform the other models of its class on the training set. Secondly, it had to be computationally efficient. We arbitrarily set a balance between both parameters. In fact, a model that outperformed another was not chosen if the difference in performance was too small as compared to the computational time required. We used logistic regression to stack the predictions of both models. The parameters also had to be tuned for this model and this is performed similarly as for the other models. The test set used to evaluate the base models and the one used to evaluate the stacked models is the same and hence the results are comparable. This method requires predictions (probabilities) to be made for the tuning set. Practically, this was done with 4-fold cross-validation. One of the folds is left out and the base models are fit on the 3 other ones. This is repeated until predictions are made for all the tuning set. The probabilities for the observations in the test set are predicted as usual. Finally, the model used for stacking (LR) is fit on the predictions for the tuning set (after cross-validation to determine the optimal parameters) and final predictions are made on the test set.

4.6. Combination of the predictions of the classification schemes

In this section, we present the method that we set up to combine the predictions of each layer of a classification scheme and how we determined the top N predictions. The top predictions were analyzed for each label separately and for the labels combined (general prediction).

The final predictions for each of the 10 labels were computed by multiplying the probabilities obtained in the first layer with their corresponding label in the second layer. For instance, the probability for an observation to be active during the exponential phase was multiplied with

the probability to interact with σ^x during the exponential phase. Then, the top N predictions (promoters) for each label were taken separately to plot the precision-top N curve. The top predictions are the promoters for which the final probability to interact with a given σ factor during a certain phase is the highest. Hence, there is no need for assigning a threshold.

The approach that was used to extract the top N general predictions is different. Indeed, the promoters in the top predictions do not necessarily interact with all the σ factors during each phase. Moreover, the certainty on each label is not necessarily the highest one. However, the certainty across the labels is in generally high. As negative labels (0) could also be present in the top predictions, the assignment of a threshold on the predictions was required. The thresholds for each label were applied separately in each layer. Afterwards, the final labels of each observation were computed by multiplying the labels in the first layer with their corresponding labels in the second layer. Hence, both layers required to agree on the assignment of a positive label to make the final label positive. Afterwards, the certainty on the whole labels that were assigned to an observation was computed. This was done by multiplying the certainty across the labels for each observation. The certainty for a label is given by Eq. 12.

$$\begin{cases} p_{final}(label) & if\ label = 1 \\ 1 - p_{final}(label) & if\ label = 0 \end{cases} \quad (12)$$

A limitation for this method is that the first layer may greatly influence the final label. In the phase-σ scheme, one label of the first layer influences half of the predictions in the second layer. In the σ-phase scheme, a prediction in the first layer (σ^x) influences only 2 labels in the second layer. If the probability for a label is below the threshold in the first layer, all the labels depending on that label will be set to 0, even if the certitude on a label was high in the second layer. The error made in the first layer is transferred to the second layer.

The threshold that was assigned to the predicted probabilities in each layer was 0.5. Indeed, the optimal thresholds that were computed for the comparison with the BacPP tool never lay far from 0.5 [0.48, 0.52]. Next to that, we thought that selecting the top predictions automatically picked up observations that had overall probabilities closer from [0, 1] in each layer. However, there is a risk that some of the labels for which there is not much certitude are then incorrectly classified in the top predictions. Thus, this method can be discussed and is likely not be the optimal one. However, we decided not to investigate computationally on other results that may arise using a different threshold for each label. As a further research, we thought of a method that may be more reliable for estimating the precision of the top N predictions across all the labels. Instead of optimizing the thresholds for the predictions in each layer, we could simply take the final probability and determine the optimal threshold on the 10 labels afterwards.

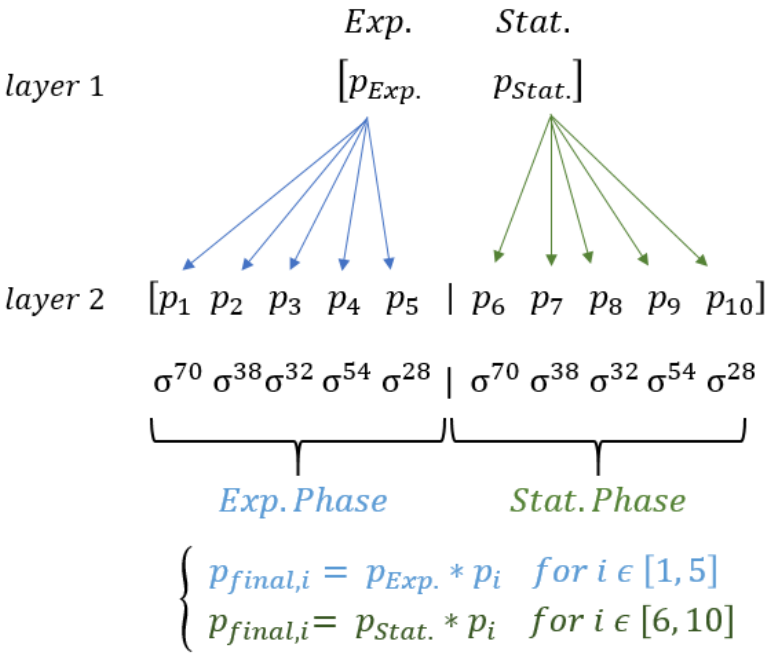


Figure 34. Combination of the predictions for the phase-σ scheme.

Alberts B, Lewis J, Johnson A, Morgan D, Raff M, Roberts K & Walter P (2015) Molecular Biology of the Cell 6th ed. Alberts B Johnson A Lewis J Morgan D Raff M Roberts K & Walter P (eds) Garland Science

Andersen KB & Von Meyenburg K (1980) Are growth rates of Escherichia coli in batch cultures limited by respiration? *J. Bacteriol.* **144:** 114–123 Available at: http://www.ncbi.nlm.nih.gov/pubmed/6998942 [Accessed May 1, 2018]

Annala M, Laurila K, Lähdesmäki H & Nykter M (2011) A Linear Model for Transcription Factor Binding Affinity Prediction in Protein Binding Microarrays. *PLoS One* **6:** e20059 Available at: http://dx.plos.org/10.1371/journal.pone.0020059 [Accessed May 16, 2018]

Artsimovitch I, Patlan V, Sekine SI, Vassylyeva MN, Hosaka T, Ochi K, Yokoyama S & Vassylyev DG (2004) Structural basis for transcription regulation by alarmone ppGpp. *Cell* **117:** 299–310 Available at: http://www.ncbi.nlm.nih.gov/pubmed/15109491 [Accessed May 1, 2018]

Assunção MD, Calheiros RN, Bianchi S, Netto MAS & Buyya R (2015) Big Data computing and clouds: Trends and future directions. *J. Parallel Distrib. Comput.* **79–80:** 3–15 Available at: https://www.sciencedirect.com/science/article/pii/S0743731514001452 [Accessed May 10, 2018]

de Avila e Silva S, Echeverrigaray S & Gerhardt GJL (2011) BacPP: Bacterial promoter prediction—A tool for accurate sigma-factor specific assignment in enterobacteria. *J. Theor. Biol.* **287:** 92–99 Available at: https://www.sciencedirect.com/science/article/pii/S0022519311003675?via%3Dihub [Accessed May 13, 2018]

Babu MM, Luscombe NM, Aravind L, Gerstein M & Teichmann SA (2004) Structure and evolution of transcriptional regulatory networks. *Curr. Opin. Struct. Biol.* **14:** 283–291 Available at: https://www.sciencedirect.com/science/article/pii/S0959440X04000788 [Accessed May 17, 2018]

Babyak MA (2004) What you see may not be what you get: A brief, nontechnical introduction to overfitting in regression-type models. *Psychosom. Med.* **66:** 411–421

Bar-Nahum G & Nudler E (2001) Isolation and characterization of sigma(70)-retaining transcription elongation complexes from Escherichia coli. *Cell* **106:** 443–51 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11525730 [Accessed February 22, 2018]

Barrick JE, Sudarsan N, Weinberg Z, Ruzzo WL & Breaker RR (2005) 6S RNA is a widespread regulator of eubacterial RNA polymerase that resembles an open promoter. *RNA* **11:** 774–784 Available at: http://www.ncbi.nlm.nih.gov/pubmed/15811922 [Accessed May 3, 2018]

Benos P V, Bulyk ML & Stormo GD (2002) Additivity in protein-DNA interactions: how good an approximation is it? *Nucleic Acids Res.* **30:** 4442–51 Available at: http://www.ncbi.nlm.nih.gov/pubmed/12384591 [Accessed May 17, 2018]

Berg J, Tymoczko J & Stryer L (2012) Biochemistry 7th ed. W. H. Freeman and Company Available at: http://www.researchgate.net/profile/James_Zimmerman/publication/264657044_Biochemistry/links/54cd5dca0cf298d6565d5962.pdf

Bernardo LMD, Johansson LUM, Solera D, Skarfstad E & Shingler V (2006) The guanosine tetraphosphate (ppGpp) alarmone, DksA and promoter affinity for RNA polymerase in regulation of sigma54-dependent transcription. *Mol. Microbiol.* **60:** 749–764 Available at: http://www.ncbi.nlm.nih.gov/pubmed/16629675 [Accessed April 27, 2018]

Bernstein E & Allis CD (2005) RNA meets chromatin. *Genes Dev.* **19:** 1635–55 Available at: http://www.ncbi.nlm.nih.gov/pubmed/16024654 [Accessed May 18, 2018]

Brennan CA, Dombroski AJ & Platt T (1987) Transcription termination factor rho is an RNA-DNA helicase. *Cell* **48:** 945–952 Available at: http://www.ncbi.nlm.nih.gov/pubmed/3030561 [Accessed April 10, 2018]

Browning DF & Busby SJW (2004) The regulation of bacterial transcription initiation. *Nat. Rev. Microbiol.* **2:** 57–65 Available at: http://www.ncbi.nlm.nih.gov/pubmed/15035009 [Accessed February 16, 2018]

Bulyk ML, Johnson PLF & Church GM (2002) Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors. *Nucleic Acids Res.* **30:** 1255–61 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11861919 [Accessed May 17, 2018]

Cai Y & Sun Y (2011) ESPRIT-Tree: Hierarchical clustering analysis of millions of 16S rRNA pyrosequences in

quasilinear computational time. *Nucleic Acids Res.* **39:** e95 Available at: http://www.ncbi.nlm.nih.gov/pubmed/21596775 [Accessed May 7, 2018]

Carbonell I (2016) The Ethics of Big Data in Big Agriculture. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2772247 [Accessed May 10, 2018]

Cavanagh AT, Sperger JM & Wassarman KM (2012) Regulation of 6S RNA by pRNA synthesis is required for efficient recovery from stationary phase in E. coli and B. subtilis. *Nucleic Acids Res.* **40:** 2234–2246 Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkr1003 [Accessed May 3, 2018]

Chen J, Wassarman KM, Feng S, Leon K, Feklistov A, Winkelman JT, Li Z, Walz T, Campbell EA & Darst SA (2017) 6S RNA Mimics B-Form DNA to Regulate Escherichia coli RNA Polymerase. *Mol. Cell* **68:** 388–397.e6 Available at: https://www.sciencedirect.com/science/article/pii/S1097276517306548 [Accessed May 3, 2018]

Chen M, Mao S & Liu Y (2014a) Big Data: A Survey. *Mob. Networks Appl.* **19:** 171–209 Available at: http://link.springer.com/10.1007/s11036-013-0489-0 [Accessed May 10, 2018]

Chen W, Lei T-Y, Jin D-C, Lin H & Chou K-C (2014b) PseKNC: A flexible web server for generating pseudo K-tuple nucleotide composition. *Anal. Biochem.* **456:** 53–60 Available at: https://www.sciencedirect.com/science/article/pii/S0003269714001249 [Accessed May 18, 2018]

Cho B-K, Kim D, Knight EM, Zengler K & Palsson BO (2014) Genome-scale reconstruction of the sigma factor network in Escherichia coli: topology and functional states. *BMC Biol.* **12:** 4 Available at: http://www.ncbi.nlm.nih.gov/pubmed/24461193 [Accessed February 16, 2018]

Cliften P, Sudarsanam P, Desikan A, Fulton L, Fulton B, Majors J, Waterston R, Cohen BA & Johnston M (2003) Finding functional features in Saccharomyces genomes by phylogenetic footprinting. *Science (80-.).* **301:** 71–76

Crooks GE, Hon G, Chandonia JM & Brenner SE (2004) WebLogo: A sequence logo generator. *Genome Res.* **14:** 1188–1190 Available at: http://www.genome.org/cgi/doi/10.1101/ [Accessed June 11, 2018]

Dove SL, Darst SA & Hochschild A (2003) Region 4 of σ as a target for transcription regulation. *Mol. Microbiol.* **48:** 863–874 Available at: http://doi.wiley.com/10.1046/j.1365-2958.2003.03467.x [Accessed February 22, 2018]

Down TA, Bergman CM, Su J & Hubbard TJP (2007) Large-Scale Discovery of Promoter Motifs in Drosophila melanogaster. *PLoS Comput. Biol.* **3:** e7 Available at: http://dx.plos.org/10.1371/journal.pcbi.0030007 [Accessed May 11, 2018]

Epshtein V, Dutta D, Wade J & Nudler E (2010) An allosteric mechanism of Rho-dependent transcription termination. *Nature* **463:** 245–249 Available at: http://www.ncbi.nlm.nih.gov/pubmed/20075920 [Accessed April 10, 2018]

Farnham PJ & Platt T (1980) A model for transcription termination suggested by studies on the trp attenuator in vitro using base analogs. *Cell* **20:** 739–48 Available at: http://www.ncbi.nlm.nih.gov/pubmed/6998564 [Accessed April 10, 2018]

Finn RD, Orlova E V, Gowen B, Buck M & Van Heel M (2000) Escherichia coli RNA polymerase core and holoenzyme structures. *EMBO J.* **19:** 6833–6844 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11118218 [Accessed February 15, 2018]

Foat BC, Morozov A V. & Bussemaker HJ (2006) Statistical mechanical modeling of genome-wide transcription factor occupancy data by MatrixREDUCE. *Bioinformatics* **22:** e141–e149 Available at: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/btl223 [Accessed May 16, 2018]

Glyde R, Ye F, Darbari VC, Zhang N, Buck M & Zhang X (2017) Structures of RNA Polymerase Closed and Intermediate Complexes Reveal Mechanisms of DNA Opening and Transcription Initiation. *Mol. Cell* **67:** 106–116.e4 Available at: http://www.ncbi.nlm.nih.gov/pubmed/28579332 [Accessed February 26, 2018]

Goñi JR, Pérez A, Torrents D & Orozco M (2007) Determining promoter location based on DNA structure first-principles calculations. *Genome Biol.* **8:** R263 Available at: http://genomebiology.biomedcentral.com/articles/10.1186/gb-2007-8-12-r263 [Accessed May 15, 2018]

Gourse RL, Ross W & Gaal T (2000) UPs and downs in bacterial transcription initiation: the role of the alpha subunit of RNA polymerase in promoter recognition. *Mol. Microbiol.* **37:** 687–95 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10972792 [Accessed February 15, 2018]

Gunnelius L, Hakkila K, Kurkela J, Wada H, Tyystjärvi E & Tyystjärvi T (2014) The omega subunit of the RNA polymerase core directs transcription efficiency in cyanobacteria. *Nucleic Acids Res.* **42:** 4606–4614 Available at: http://www.ncbi.nlm.nih.gov/pubmed/24476911 [Accessed February 16, 2018]

Hastie T, Tibshirani R & Friedman J (2009) Unsupervised Learning. In *The Elements of Statistical Learning* pp 485–585. Springer, New York, NY Available at: http://link.springer.com/10.1007/978-0-387-84858-7_14 [Accessed May 10, 2018]

Jacobson Ralph (2013) 2.5 quintillion bytes of data created every day. How does CPG & Retail manage it? - IBM Consumer Products Industry Blog. Available at: https://www.ibm.com/blogs/insights-on-business/consumer-products/2-5-quintillion-bytes-of-data-created-every-day-how-does-cpg-retail-manage-it/ [Accessed May 7, 2018]

James G, Witten D, Hastie T & Tibshirani R (2000) An introduction to Statistical Learning

Jiang M, Ma N, Vassylyev DG & McAllister WT (2004) RNA displacement and resolution of the transcription bubble during transcription by R7 RNA polymerase. *Mol. Cell* **15:** 777–788 Available at: http://www.ncbi.nlm.nih.gov/pubmed/15350221 [Accessed February 27, 2018]

Jishage M, Kvint K, Shingler V & Nyström T (2002) Regulation of σ factor competition by the alarmone ppGpp. *Genes Dev.* **16:** 1260–1270 Available at: http://www.ncbi.nlm.nih.gov/pubmed/12023304 [Accessed May 1, 2018]

Kang J, Hahn M-Y, Ishihama A & Roe J-H (1997) Identification of sigma factors for growth phase-related promoter selectivity of RNA polymerases from Streptomyces coelicolor A3(2). *Nucleic Acids Res.* **25:** 2566–2573 Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/25.13.2566 [Accessed April 27, 2018]

Kang JY, Mishanina T V., Bellecourt MJ, Mooney RA, Darst SA & Landick R (2018) RNA Polymerase Accommodates a Pause RNA Hairpin by Global Conformational Rearrangements that Prolong Pausing. *Mol. Cell* **69:** 802–815.e1 Available at: https://www.sciencedirect.com/science/article/pii/S1097276518300479 [Accessed May 21, 2018]

Kapanidis AN, Margeat E, Ho SO, Kortkhonjia E, Weiss S & Ebright RH (2006) INITIAL TRANSCRIPTION BY RNA POLYMERASE PROCEEDS THROUGH A DNA-SCRUNCHING MECHANISM: Single-molecule fluorescence-resonance-energy-transfer experiments establish that initial transcription proceeds through a 'scrunching' mechanism, in which RNA polymerase. *Science* **314:** 1144–1147 Available at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2754788/

Keuth S & Bisping B (1994) Vitamin B12 production by Citrobacter freundii or Klebsiella pneumoniae during tempeh fermentation and proof of enterotoxin absence by PCR. *Appl. Environ. Microbiol.* **60:** 1495–9 Available at: http://www.ncbi.nlm.nih.gov/pubmed/8017933 [Accessed April 26, 2018]

Kien NB, Kong I-S, Lee M-G & Kim JK (2010) Coenzyme Q10 production in a 150-l reactor by a mutant strain of Rhodobacter sphaeroides. *J. Ind. Microbiol. Biotechnol.* **37:** 521–529 Available at: http://link.springer.com/10.1007/s10295-010-0699-4 [Accessed April 26, 2018]

Lenhard B, Sandelin A, Mendoza L, Engström P, Jareborg N & Wasserman WW (2003) Identification of conserved regulatory elements by comparative genome analysis. *J. Biol.* **2:** 13 Available at: http://jbiol.biomedcentral.com/articles/10.1186/1475-4924-2-13 [Accessed May 18, 2018]

Li R, Zhu H, Ruan J, Qian W, Fang X, Shi Z, Li Y, Li S, Shan G, Kristiansen K, Li S, Yang H, Wang J & Wang J (2010) De novo assembly of human genomes with massively parallel short read sequencing. *Genome Res.* **20:** 265–72 Available at: http://www.ncbi.nlm.nih.gov/pubmed/20019144 [Accessed May 11, 2018]

Lin H, Deng EZ, Ding H, Chen W & Chou KC (2014) IPro54-PseKNC: A sequence-based predictor for identifying sigma-54 promoters in prokaryote with pseudo k-tuple nucleotide composition. *Nucleic Acids Res.* **42:** 12961–12972 Available at: http://academic.oup.com/nar/article/42/21/12961/2902492/iPro54PseKNC-a-sequencebased-predictor-for [Accessed February 26, 2018]

Ma S, Shah S, Bohnert HJ, Snyder M & Dinesh-Kumar SP (2013) Incorporating motif analysis into gene co-expression networks reveals novel modular expression pattern and new signaling pathways. *PLoS Genet.* **9:** e1003840 Available at: http://www.ncbi.nlm.nih.gov/pubmed/24098147 [Accessed May 18, 2018]

Van Der Maaten L & Hinton G (2008) Visualizing Data using t-SNE. *J. Mach. Learn. Res.* **9:** 2579–2605 Available at: http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf [Accessed June 8, 2018]

Maeda H (2000) Competition among seven Escherichia coli sigma subunits: relative binding affinities to the core RNA polymerase. *Nucleic Acids Res.* **28:** 3497–3503 Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/28.18.3497 [Accessed April 9, 2018]

Monsion B, Incarbone M, Hleibieh K, Poignavent V, Ghannam A, Dunoyer P, Daeffler L, Tilsner J & Ritzenthaler C (2018) Efficient Detection of Long dsRNA in Vitro and in Vivo Using the dsRNA Binding Domain from FHV B2 Protein. *Front. Plant Sci.* **9:** 70 Available at: http://journal.frontiersin.org/article/10.3389/fpls.2018.00070/full [Accessed May 21, 2018]

Mukhopadhyay J, Kapanidis AN, Mekler V, Kortkhonjia E, Ebright YW & Ebright RH (2001) Translocation of sigma(70) with RNA polymerase during transcription: fluorescence resonance energy transfer assay for movement relative to DNA. *Cell* **106:** 453–63 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11525731 [Accessed February 22, 2018]

Murakami KS (2015) Structural Biology of Bacterial RNA Polymerase. *Biomolecules* **5:** 848–864 Available at: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4496699/

Nagai H & Shimamoto N (1997) Regions of the Escherichia coli primary sigma factor sigma70 that are involved in interaction with RNA polymerase core enzyme. *Genes Cells* **2:** 725–34 Available at: http://www.ncbi.nlm.nih.gov/pubmed/9544701 [Accessed February 22, 2018]

Nandy Mazumdar M, Nedialkov Y, Svetlov D, Sevostyanova A, Belogurov GA & Artsimovitch I (2016) RNA polymerase gate loop guides the nontemplate DNA strand in transcription complexes. *Proc. Natl. Acad. Sci.* **113:** 14994–14999 Available at: http://www.ncbi.nlm.nih.gov/pubmed/27956639 [Accessed February 22, 2018]

Nelson JD, Denisenko O, Sova P & Bomsztyk K (2006) Fast chromatin immunoprecipitation assay. *Nucleic Acids Res.* **34:** e2–e2 Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gnj004 [Accessed May 18, 2018]

Paget MS (2015) Bacterial Sigma Factors and Anti-Sigma Factors: Structure, Function and Distribution. *Biomolecules* **5:** 1245–65 Available at: http://www.ncbi.nlm.nih.gov/pubmed/26131973 [Accessed February 16, 2018]

Perederina A, Svetlov V, Vassylyeva MN, Tahirov TH, Yokoyama S, Artsimovitch I & Vassylyev DG (2004) Regulation through the secondary channel - Structural framework for ppGpp-DksA synergism during transcription. *Cell* **118:** 297–309 Available at: http://www.ncbi.nlm.nih.gov/pubmed/15294156 [Accessed May 1, 2018]

Pletnev P, Osterman I, Sergiev P, Bogdanov A & Dontsova O (2015) Survival guide: Escherichia coli in the stationary phase. *Acta Naturae* **7:** 22–33 Available at: http://www.ncbi.nlm.nih.gov/pubmed/26798489 [Accessed April 26, 2018]

Raffaelle M, Kanin EI, Vogt J, Burgess RR & Ansari AZ (2005) Holoenzyme Switching and Stochastic Release of Sigma Factors from RNA Polymerase In Vivo. *Mol. Cell* **20:** 357–366 Available at: https://www.sciencedirect.com/science/article/pii/S1097276505016813 [Accessed May 19, 2018]

Ratsch G, Sonnenburg S & Scholkopf B (2005) RASE: recognition of alternatively spliced exons in C.elegans. *Bioinformatics* **21:** i369–i377 Available at: https://academic.oup.com/bioinformatics/article-lookup/doi/10.1093/bioinformatics/bti1053 [Accessed June 8, 2018]

Read JE (2017) Chromatin Immunoprecipitation and Quantitative Real-Time PCR to Assess Binding of a Protein of Interest to Identified Predicted Binding Sites Within a Promoter. In *Methods in molecular biology (Clifton, N.J.)* pp 23–32. Available at: http://www.ncbi.nlm.nih.gov/pubmed/28801897 [Accessed May 18, 2018]

Reddy G, Altaf M, Naveena BJ, Venkateshwar M & Kumar EV (2008) Amylolytic bacterial lactic acid fermentation

— A review. *Biotechnol. Adv.* **26:** 22–34 Available at: https://www.sciencedirect.com/science/article/pii/S0734975007000961 [Accessed April 26, 2018]

Riley TR, Lazarovici A, Mann RS & Bussemaker HJ (2015) Building accurate sequence-to-affinity models from high-throughput in vitro protein-DNA binding data using FeatureREDUCE. *Elife* **4:** Available at: http://www.ncbi.nlm.nih.gov/pubmed/26701911 [Accessed May 16, 2018]

Roberts JW (1969) Termination factor for RNA synthesis. *Nature* **224:** 1168–1174 Available at: http://www.nature.com/doifinder/10.1038/2241168a0 [Accessed April 10, 2018]

Rodionov DA (2007) Comparative Genomic Reconstruction of Transcription Regulatory Networks in Bacteria. *Chem. Rev.* **107:** 3467–3497 Available at: https://pubs.acs.org/doi/abs/10.1021/cr068309+ [Accessed May 18, 2018]

Rolfe MD, Rice CJ, Lucchini S, Pin C, Thompson A, Cameron ADS, Alston M, Stringer MF, Betts RP, Baranyi J, Peck MW & Hinton JCD (2012) Lag phase is a distinct growth phase that prepares bacteria for exponential growth and involves transient metal accumulation. *J. Bacteriol.* **194:** 686–701 Available at: http://www.ncbi.nlm.nih.gov/pubmed/22139505 [Accessed April 26, 2018]

Ross W, Ernst A & Gourse RL (2001) Fine structure of E. coli RNA polymerase-promoter interactions: α subunit binding to the UP element minor groove. *Genes Dev.* **15:** 491–506 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11238372 [Accessed February 20, 2018]

Saecker RM, Record MT, Dehaseth PL & deHaseth PL (2011) Mechanism of bacterial transcription initiation: RNA polymerase - promoter binding, isomerization to initiation-competent open complexes, and initiation of RNA synthesis. *J. Mol. Biol.* **412:** 754–71 Available at: http://www.ncbi.nlm.nih.gov/pubmed/21371479 [Accessed February 22, 2018]

San K -Y & Stephanopoulos G (1989) Optimization of fed-batch penicillin fermentation: A case of singular optimal control with state constraints. *Biotechnol. Bioeng.* **34:** 72–78 Available at: http://doi.wiley.com/10.1002/bit.260340110 [Accessed April 26, 2018]

Sanderson A, Mitchell JE, Minchin SD & Busby SJ. (2003) Substitutions in the Escherichia coli RNA polymerase σ70factor that affect recognition of extended -10 elements at promoters. *FEBS Lett.* **544:** 199–205 Available at: https://www.sciencedirect.com/science/article/pii/S0014579303005003 [Accessed February 20, 2018]

Scherf M, Klingenhoff A & Werner T (2000) Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach. *J. Mol. Biol.* **297:** 599–606 Available at: https://www.sciencedirect.com/science/article/pii/S0022283600935897 [Accessed May 13, 2018]

Shimada T, Tanaka K & Ishihama A (2017) The whole set of the constitutive promoters recognized by four minor sigma subunits of Escherichia coli RNA polymerase. *PLoS One* **12:** e0179181 Available at: http://www.ncbi.nlm.nih.gov/pubmed/28666008 [Accessed February 16, 2018]

Siddharthan R (2010) Dinucleotide Weight Matrices for Predicting Transcription Factor Binding Sites: Generalizing the Position Weight Matrix. *PLoS One* **5:** e9722 Available at: http://dx.plos.org/10.1371/journal.pone.0009722 [Accessed May 16, 2018]

Steger D, Berry D, Haider S, Horn M, Wagner M, Stocker R & Loy A (2011) Systematic Spatial Bias in DNA Microarray Hybridization Is Caused by Probe Spot Position-Dependent Variability in Lateral Diffusion. *PLoS One* **6:** e23727 Available at: http://www.ncbi.nlm.nih.gov/pubmed/21858215 [Accessed May 11, 2018]

Structure and Function of DNA (2016) *web page* Available at: https://courses.lumenlearning.com/microbiology/chapter/structure-and-function-of-dna/ [Accessed May 21, 2018]

Toulokhonov I, Artsimovitch I & Landick R (2001) Allosteric control of RNA polymerase by a site that contacts nascent RNA hairpins. *Science (80-.).* **292:** 730–733 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11326100 [Accessed March 1, 2018]

Typas A, Barembruch C, Possling A & Hengge R (2007) Stationary phase reorganisation of the Escherichia coli transcription machinery by Crl protein, a fine-tuner of sigmas activity and levels. *EMBO J.* **26:** 1569–78 Available at: http://www.ncbi.nlm.nih.gov/pubmed/17332743 [Accessed April 27, 2018]

Valouev A, Johnson DS, Sundquist A, Medina C, Anton E, Batzoglou S, Myers RM & Sidow A (2008) Genome-wide

analysis of transcription factor binding sites based on ChIP-Seq data. *Nat. Methods* **5:** 829–834 Available at: http://www.nature.com/articles/nmeth.1246 [Accessed May 11, 2018]

Vassylyev DG, Vassylyeva MN, Perederina A, Tahirov TH & Artsimovitch I (2007) Structural basis for transcription elongation by bacterial RNA polymerase. *Nature* **448:** 157–162 Available at: http://www.nature.com/articles/nature05932 [Accessed May 21, 2018]

Vuthoori S, Bowers CW, McCracken A, Dombroski AJ & Hinton DM (2001) Domain 1.1 of the σ70subunit of Escherichia coli RNA polymerase modulates the formation of stable polymerase/promoter complexes. *J. Mol. Biol.* **309:** 561–572 Available at: http://www.ncbi.nlm.nih.gov/pubmed/11397080 [Accessed February 20, 2018]

Wassarman KM & Storz G (2000) 6S RNA regulates E. coli RNA polymerase activity. *Cell* **101:** 613–23 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10892648 [Accessed April 27, 2018]

Weaver R (2011) Molecular Biology 5th ed. McGraw-Hill Available at: http://doi.wiley.com/10.1002/bmb.8

Weirauch MT, Cote A, Norel R, Annala M, Zhao Y, Riley TR, Saez-Rodriguez J, Cokelaer T, Vedenko A, Talukder S, Bussemaker HJ, Morris QD, Bulyk ML, Stolovitzky G, Hughes TR, Chang CW, Chen C-Y, Chen Y-S, Chu Y-W, Grau J, et al (2013) Evaluation of methods for modeling transcription factor sequence specificity. *Nat. Biotechnol.* **31:** 126–134 Available at: http://www.nature.com/articles/nbt.2486 [Accessed May 17, 2018]

Weixlbaumer A, Leon K, Landick R & Darst SA (2013) Structural basis of transcriptional pausing in bacteria. *Cell* **152:** 431–41 Available at: http://www.ncbi.nlm.nih.gov/pubmed/23374340 [Accessed March 1, 2018]

Wu X & Bartel DP (2017) kpLogo: positional k-mer analysis reveals hidden specificity in biological sequences. *Nucleic Acids Res.* **45:** W534–W538 Available at: https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gkx323 [Accessed May 17, 2018]

Xia X (2012) Position weight matrix, gibbs sampler, and the associated significance tests in motif characterization and prediction. *Scientifica (Cairo).* **2012:** 917540 Available at: http://www.ncbi.nlm.nih.gov/pubmed/24278755 [Accessed May 17, 2018]

Yakhnin A V & Babitzke P (2010) Mechanism of NusG-stimulated pausing, hairpin-dependent pause site selection and intrinsic termination at overlapping pause and termination sites in the Bacillus subtilis trp leader. *Mol. Microbiol.* **76:** 690–705 Available at: http://www.ncbi.nlm.nih.gov/pubmed/20384694 [Accessed March 1, 2018]

Yarnell WS & Roberts JW (1999) Mechanism of intrinsic transcription termination and antitermination. *Science (80-.).* **284:** 611–615 Available at: http://www.ncbi.nlm.nih.gov/pubmed/10213678 [Accessed April 10, 2018]

Zaslaver A, Kaplan S, Bren A, Jinich A, Mayo A, Dekel E, Alon U & Itzkovitz S (2009) Invariant Distribution of Promoter Activities in Escherichia coli. *PLoS Comput. Biol.* **5:** e1000545 Available at: http://dx.plos.org/10.1371/journal.pcbi.1000545 [Accessed April 30, 2018]

APPENDICES

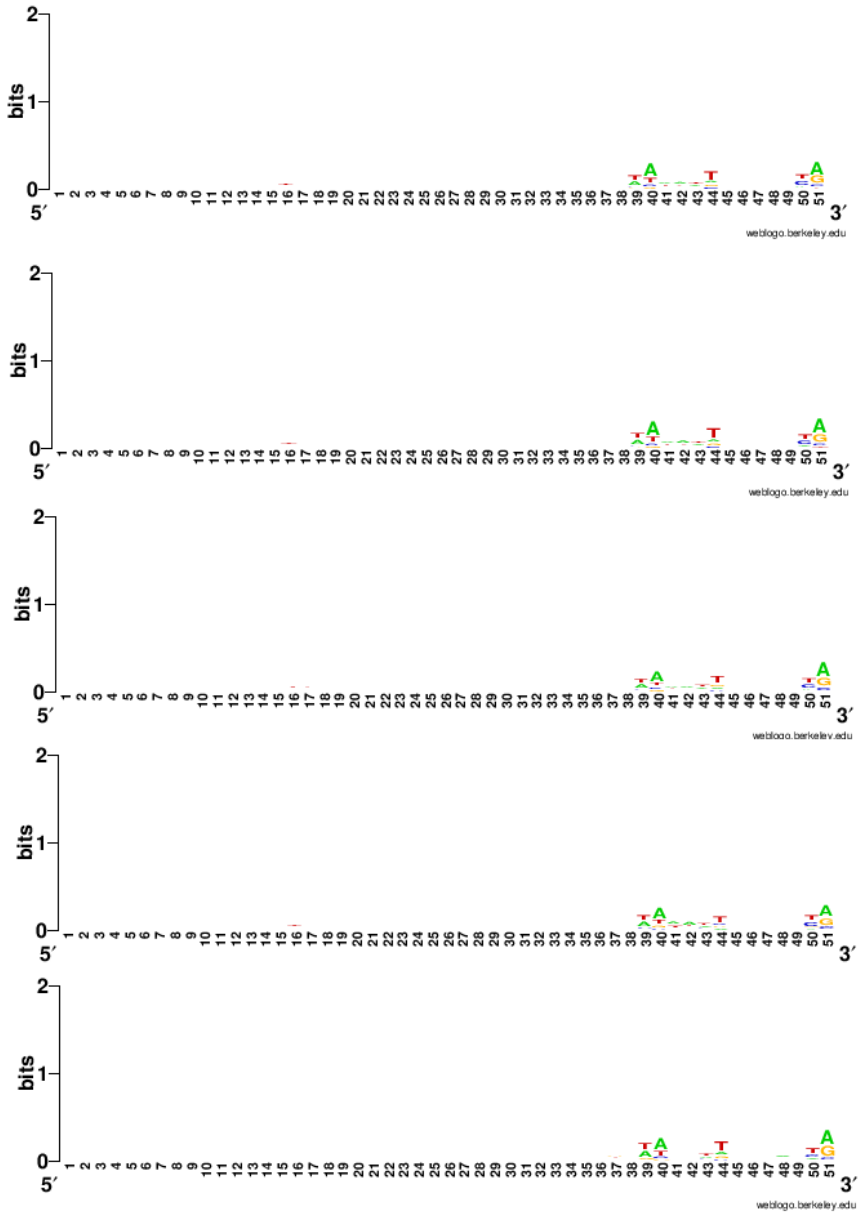
Appendix 1. Number of overlapping promoters between σ-factors for 3 different conditions.

Grouped phases	S70	S38	S32	S54	S28
S70	3299				
S38	1986	2120			
S32	726	539	783		
S54	342	265	83	370	
S28	116	99	22	24	123

Exp. phase	S70	S38	S32	S54	S28
S70	1808				
S38	1091	1161			
S32	385	278	413		
S54	169	137	33	187	
S28	64	57	11	13	67

Stat. phase	S70	S38	S32	S54	S28
S70	2364				
S38	1420	1520			
S32	513	389	560		
S54	259	201	69	279	
S28	75	62	11	14	81

Appendix 2. Overrepresented motifs. From top to bottom: σ^{70}, σ^{38}, σ^{32}, σ^{54} and σ^{28} promoters during the exponential phase. (Crooks *et al*, 2004)



Appendix 3 Venn diagrams for the overlap between promoters recognized by certain σ factors. From left to right: promoters active when phases are grouped and during the exponential phase. Bottom: Promoters active during the stationary phase.