

METHODOLOGIES FOR PREDICTION OF TRANSCRIPTION FACTORS IN
TRANSCRIPTIONAL REGULATORY MECHANISMS IN BIOCATALYSIS OF
REACTIONS IN YEAST CENTRAL PATHWAYS

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

OĞUZ ULAŞ YAMAN

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
CHEMICAL ENGINEERING

AUGUST 2022

Approval of the thesis:

**METHODOLOGIES FOR PREDICTION OF TRANSCRIPTION FACTORS
IN TRANSCRIPTIONAL REGULATORY MECHANISMS IN
BIOCATALYSIS OF REACTIONS IN YEAST CENTRAL PATHWAYS**

submitted by **OĞUZ ULAŞ YAMAN** in partial fulfillment of the requirements for
the degree of **Master of Science in Chemical Engineering Department, Middle
East Technical University** by,

Prof. Dr. Halil Kalıpçılar
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Pınar Çalık
Head of Department, **Chemical Engineering**

Prof. Dr. Pınar Çalık
Supervisor, **Chemical Engineering, METU**

Examining Committee Members:

Prof. Dr. Tunçer H. Özdamar
Chemical Engineering, Ankara University

Prof. Dr. Pınar Çalık
Chemical Engineering, METU

Assoc. Prof. Dr. Harun Koku
Chemical Engineering, METU

Assist. Prof. Dr. Gökhan Çelik
Chemical Engineering, METU

Assist. Prof. Dr. Aybar Can Acar
Bioinformatics, METU

Date:



I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Surname: Oğuz Ulaş Yaman

Signature :

ABSTRACT

METHODOLOGIES FOR PREDICTION OF TRANSCRIPTION FACTORS IN TRANSCRIPTIONAL REGULATORY MECHANISMS IN BIOCATALYSIS OF REACTIONS IN YEAST CENTRAL PATHWAYS

Yaman, Oğuz Ulaş

M.S., Department of Chemical Engineering

Supervisor: Prof. Dr. Pınar Çalık

August 2022, 143 pages

In this MSc thesis, the aim is to propose methodologies for predicting transcription factor binding sites in yeast cells. This aim is achieved, first, by modeling *P. pastoris* central carbon metabolism genes using phylogenetic footprinting; and next, by modelling *S. cerevisiae* transcription factors' affinity towards 8-mers using the Machine Learning algorithmic models, i.e., Random Forest, XGBoost, and Deep Learning.

In the first part of the thesis, a novel phylogenetic footprinting algorithm is introduced, which requires any number of orthologous promoter pairs with their DNA sequences, and a database that contains the transcription factor binding motifs as input. The model first scans the reference promoter for TF binding sites, and then using pairwise alignment, determines the conserved transcription factor binding sites in the target promoter. The algorithm was used to compare 58 *S. cerevisiae* promoters of the genes in the central carbon metabolism with the predicted 52 orthologous *P. pastoris* promoters. The presented phylogenetic footprinting predictions of transcription factor binding sites enabled annotation of 116 *P. pastoris* transcription factors in the central pathways.

In the second part of the thesis, seven Machine Learning algorithmic models (five based on Neural Networks, one based on XGBoost, and one based on Random Forest) were trained to predict high affinity 8-mers for *S. cerevisiae* transcription factors. The 8-mers were represented embedded into numerical arrays with using the predetermined five features that can represent sequence specificities of the transcription factor binding sites. Since different transcription factors may recognize different features, A greedy approach was designed, which selectively picks the best pool of features and makes the model combination for each transcription factor that gives the best Matthews Correlation Coefficient (MCC) score on test data. The presented novel approach yielded an average MCC score of 0.873 in predicting high-affinity binding sites for all the transcription factors.

Keywords: Transcription Factor, Transcription Factor Binding Site Prediction, Yeast, Phylogenetic Footprinting, Machine Learning

ÖZ

MAYALARIN MERKEZİ TEPKİME YOLİZLERİNDEKİ REAKSİYONLARIN BİYOKATALİZİNDE TRANSKRİPSİYONEL REGÜLASYON MEKANİZMALARINDAKİ TRANSKRİPSİYON FAKTÖRLERİNİN TAHMİNLENMESİ İÇİN METODOLOJİLER

Yaman, Oğuz Ulaş

Yüksek Lisans, Kimya Mühendisliği Bölümü

Tez Yöneticisi: Prof. Dr. Pınar Çalık

Ağustos 2022 , 143 sayfa

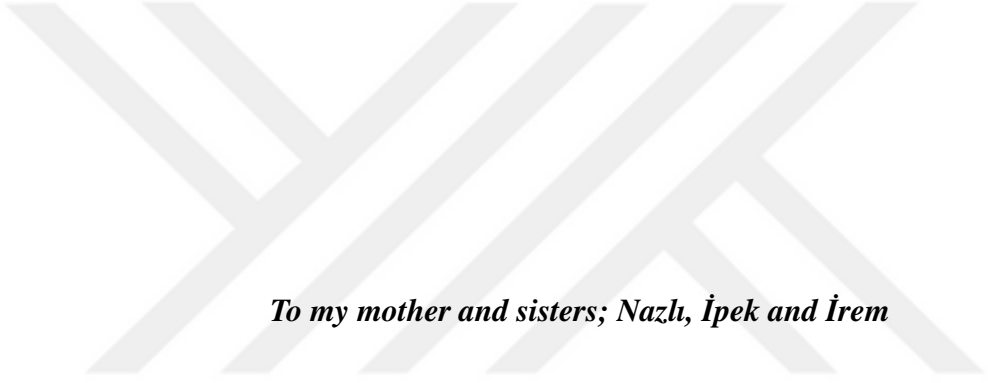
Bu tezde amaç, maya hücrelerindeki transkripsiyon faktörü bağlanma konumlarını tahmin etmek için metodolojiler önermektir. Bu amaca ulaşmak için, ilk olarak *P. pastoris* merkezi karbon metabolizmasındaki reaksiyonları katalizleyen enzimlerin genlerinin promotorlar üzerindeki bağlanma bölgeleri filogenetik ayakizi kullanarak modellenmiştir. İkinci olarak, *S. cerevisiae* transkripsiyon faktörlerinin 8-merlere bağlanma afinitesi çeşitli makine öğrenme algoritmaları, Rastgele Orman, XGBoost ve Derin Öğrenme, kullanarak modellenmiştir.

Tezde önce, DNA dizileriyle birlikte herhangi bir sayıda ortolog promotor çifti ve transkripsiyon faktörü bağlanma motiflerini içeren veri tabanını girdi olarak gerektiren bir filogenetik ayakizi algoritması betimlenmiştir. Model ilk önce referans promotoru bağlanma bölgeleri için tarama yapar, ardından ikili dizi hizalamayı kullanarak hedef promotorda korunmuş transkripsiyon faktörü bağlanma bölgelerini belirler. Algoritma merkezi karbon metabolizması üzerindeki tepkimeleri katalizleyen enzim-

lerin genleri için 58 *S. cerevisiae* ve onlara karşılık belirlenen 52 ortolog *P. pastoris* promotorlarının karşılaştırmasını yapmıştır. Transkripsiyon faktörleri bağlanma konumlarının filogenetik ayakizi tahminleri, *P. pastoris* merkezi yolizlerindeki transkripsiyon faktörlerini tahminlenmesine olanak vermiştir.

Tezin ikinci bölümünde, yedi makine öğrenme algoritmik modeli (beş Yapay Sinir Ağları, bir XGBoost ve bir Rastgele Orman), *S. cerevisiae* transkripsiyon faktörleri için yüksek afiniteli (ilk %1) 8-merleri tahminlemek için eğitilmiştir. 8-merler, belirlenen 5 özellik ile sayısal dizilere gömülü temsil edilmiştir. Farklı transkripsiyon faktörleri farklı özellikleri tanıyabileceğinden, en iyi Matthews Korelasyon Katsayısını (MCC) veren, her bir transkripsiyon faktörü için en iyi özellik havuzu, model kombinasyonunu seçecek açgözlü bir yaklaşım benimsenmiştir. Böylece, tüm transkripsiyon faktörleri üzerinde yüksek afiniteli bağlanma bölgeleri ortalama 0.873 MCC skoruyla tahminlenmiştir.

Anahtar Kelimeler: Transkripsiyon Faktörü, Transkripsiyon Faktörü Bağlanma Bölgesi Tahminleme, Maya, Filogenetik Ayakizi, Makine Öğrenmesi



To my mother and sisters; Nazlı, İpek and İrem

ACKNOWLEDGMENTS

I am eternally grateful to my supervisor, Professor Dr. Pınar Çalık, who enlightened my academic path with endless support, wise advice, and great encouragement that remarkably influenced my academic and social life. I shall never forget the things I have learned from her.

I thank Assist. Professor Dr. Aybar C. Acar for giving me the opportunity to use Bioinformatics servers. I also thank Professor Halit Oğuztüzün and Dr. Özgür Kaya for giving me the opportunity to use the METU Computer Engineering Department server.

I would like to thank my labmates, Beste Avcı and Beste Avcı in Industrial Biotechnology and Bioinformatics Laboratory, Ahmet Samet Özdilek and Arda Eskin from Bioinformatics Department, and Toprak Çağlar and Öznur Doğan of Chemical Engineering Department for their companionship and supportive behavior. Especially, I would like to express my gratitude to Ahmet Samet Özdilek, İdil Dayankaç, and Öznur Doğan, whose endless support in various aspects was essential for my progress. I also shall thank Rümeyza Üstüncan for her outstanding support. I am also grateful to all academic, administrative, and technical staff of the chemical engineering department for providing a convenient atmosphere for conducting this research.

Scientific and Technical Research Council of Turkey (TÜBİTAK-119Z435), is greatly acknowledged.

I would like to express my eternal gratitude to is my mother, Nazlı Yaman, whose endless support has allowed me to push through the challenges to become the person I am today. Finally, I would like to thank my dearest sisters, İpek and İrem, whose support, not only during the span of this study but my whole life, was, is, and always will be the primary motivation behind any success I shall ever achieve.

August 8, 2022

Oğuz Ulaş Yaman

TABLE OF CONTENTS

ABSTRACT	v
ÖZ	vii
ACKNOWLEDGMENTS	x
TABLE OF CONTENTS	xi
LIST OF TABLES	xv
LIST OF FIGURES	xix
LIST OF ALGORITHMS	xxi
LIST OF ABBREVIATIONS	xxii
CHAPTERS	
1 INTRODUCTION	1
2 BIOLOGICAL BACKGROUND	5
2.1 Transcription Factors	5
2.2 Cell Lines	7
2.2.1 <i>Saccharomyces cerevisiae</i>	7
2.2.2 <i>Pichia pastoris</i>	10
3 THEORETICAL BACKGROUND	11
3.1 Phylogenetic Footprinting	11
3.1.1 Position Weight Matrices and Motif Searching	11

3.1.2	Introduction to Phylogenetic Footprinting	14
3.1.3	Determination of Homologous Sequences	15
3.1.4	TFBS Conservation Criterion	16
3.2	Machine Learning	17
3.2.1	Transition to Machine Learning	17
3.2.2	Random Forest	18
3.2.3	Extreme Gradient Boosting	20
3.2.4	Deep Learning	24
3.2.4.1	Multilayer Perceptrons	24
3.2.4.2	Stochastic Gradient Descent	25
3.2.4.3	ADAM and ADAMax	27
3.2.4.4	Binary Cross-Entropy	30
3.2.4.5	Rectified Linear Unit	31
3.2.5	Data Features	32
3.2.5.1	k-mer Dinucleotide Frequency	32
3.2.5.2	k-Spaced Nucleotide Pair Frequency	33
3.2.5.3	Nucleotide Chemical Property	33
3.2.5.4	Pseudo-k-tuple Nucleotide Composition	34
3.2.5.5	Electron-Ion Interaction Pseudopotentials of Trinucleotide	34
4	PHYLOGENETIC FOOTPRINTING METHOD FOR PREDICTING CIS- ACTING DNA ELEMENTS	37
4.1	Introduction	37
4.2	Methods	37

4.2.1	Identification of <i>S. cerevisiae</i> and <i>P. pastoris</i> Promoters	37
4.2.2	Motif Scanning and Pairwise Alignment using Biopython . . .	38
4.2.3	Performance Analysis	40
4.3	Results	40
4.3.1	Homologous Proteins of <i>S. cerevisiae</i> and <i>P. pastoris</i>	40
4.3.2	Annotation of TFBSs in <i>S. cerevisiae</i> Promoters	41
4.3.3	Annotation of TFBSs on <i>P. pastoris</i> Promoters	42
4.3.4	Computation of <i>P. pastoris</i> PWMs	44
4.3.5	Scanning <i>P. pastoris</i> promoters for unidentified TFBSs	44
4.3.6	Comparison with the Literature	45
5	NOVEL STRATEGY DESIGNED WITH MACHINE LEARNING ALGO- RITHMIC MODELS FOR PREDICTING CIS-ACTING DNA SITES . . .	49
5.1	Introduction	49
5.2	Methods	49
5.2.1	Machine Learning Method	49
5.2.1.1	Dataset	49
5.2.1.2	Embedding	49
5.2.1.3	Model Training	51
5.2.1.4	Statistical Analysis of Models	52
5.3	Results	54
5.3.1	Neural Network Optimization	54
5.3.1.1	Optimizer Selection	54
5.3.1.2	Activation Function Selection	55

5.3.2	Training Results	55
5.3.3	Feature Selection	58
5.3.3.1	1st-Order Feature Elimination	58
5.3.3.2	<i>N</i> th-Order Feature Elimination	61
5.3.3.3	Model Stability	64
5.3.4	Word2Vec Supported Models	64
6	CONCLUSIONS	67
	REFERENCES	69
	APPENDICES	
A	PROTEIN-PROTEIN BLAST RESULTS	85
B	TFBS SEQUENCE LOGOS FOR <i>S. CEREVISIAE</i> AND <i>P. PASTORIS</i> TFS	93
C	ANNOTATED TFBSS ON <i>S. CEREVISIAE</i> AND <i>P. PASTORIS</i> PROMOT- ERS	105
D	EFFECT OF REMOVING FEATURES ON MODEL PERFORMANCES .	123

LIST OF TABLES

TABLES

Table 3.1	Chemical properties of nucleotides [1]	34
Table 3.2	Contributions of the nucleotide pairs to physical, structural properties [2]	35
Table 3.3	Electron-ion interaction pseudopotential contributions for the nucleotides [3]	36
Table 4.1	NCBI BLAST Results of <i>S. cerevisiae</i> ADH2 enzyme against <i>P. pastoris</i> <i>GS115</i> database [4]	42
Table 4.2	Information of Cat8 hits on P_{ADH2} promoter	42
Table 4.3	TFBSs counts annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on <i>S. cerevisiae</i> and <i>P. pastoris</i> promoters that regulate expression of enzymes in the Glycolysis Pathway	46
Table 5.1	Optimizer Performances upon training models for 28 TFs using ReLU activation Function	55
Table 5.2	Activation Function Performances upon training models for 28 TFs using ReLU activation Function	56
Table 5.3	Performance Metrics achieved by three Machine Learning Methods over 274 TFs	56
Table 5.4	Performance Metrics of the three subgroups of TFs.	57

Table 5.5	Change of Performance Metrics with respect to removed features . .	59
Table 5.6	Performance Metrics of the three subgroups of TFs when Feature Pool size is reduced to four	61
Table 5.7	Performance Metrics of the three subgroups of TFs when Feature Pool is varied to include best performing features	63
Table 5.8	MCC Scores of Word2Vec included models with changing training hyperparameters	65
Table 5.9	ELU 3-depth Deep Learning Models performance metrics on 150 TFs	66
Table A.1	Protein-protein BLASTs of the glycolysis enzymes of <i>S. cerevisiae</i> <i>S288c</i> against non-redundant proteins database of <i>P. pastoris GS115</i> . . .	85
Table A.2	Protein-protein BLASTs of the gluconeogenesis enzymes of <i>S. cere-</i> <i>visiae S288c</i> against non-redundant proteins database of <i>P. pastoris GS115</i>	86
Table A.3	Protein-protein BLASTs of the Pentose Phosphate Pathway enzymes of <i>S. cerevisiae S288c</i> against non-redundant proteins database of <i>P. pas-</i> <i>toris GS115</i>	87
Table A.4	Protein-protein BLASTs of the Alcoholic Fermentation enzymes of <i>S. cerevisiae S288c</i> against non-redundant proteins database of <i>P. pastoris</i> <i>GS115</i>	88
Table A.5	Protein-protein BLASTs of the TCA Cycle enzymes of <i>S. cerevisiae</i> <i>S288c</i> against non-redundant proteins database of <i>P. pastoris GS115</i> . . .	88
Table A.6	Protein-protein BLASTs of the Glyoxylate Cycle enzymes of <i>S.</i> <i>cerevisiae S288c</i> against non-redundant proteins database of <i>P. pastoris</i> <i>GS115</i>	90
Table A.7	Protein-protein BLASTs of the Ethanol Utilization Pathway en- zymes of <i>S. cerevisiae S288c</i> against non-redundant proteins database of <i>P. pastoris GS115</i>	90

Table A.8 Protein-protein BLASTs of the Glycerol Utilization Pathway enzymes of <i>S. cerevisiae</i> S288c against non-redundant proteins database of <i>P. pastoris</i> GS115	91
Table A.9 Protein-protein BLASTs of the Anaplerotic reaction enzymes of <i>S. cerevisiae</i> S288c against non-redundant proteins database of <i>P. pastoris</i> GS115	91
Table B.1 WebLogo Sequence Logos [5] of <i>S. cerevisiae</i> S288c PWMs that are extracted from TransFac [6], and computed <i>P. pastoris</i> GS115 PWMs .	93
Table C.1 Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on <i>S. cerevisiae</i> and <i>P. pastoris</i> promoters that regulate expression of enzymes that act on Glycolysis and Gluconeogenesis Pathways.	105
Table C.2 Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on <i>S. cerevisiae</i> and <i>P. pastoris</i> promoters that regulate expression of enzymes that act on Pentose Phosphate Pathway.	109
Table C.3 Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on <i>S. cerevisiae</i> and <i>P. pastoris</i> promoters that regulate expression of enzymes that act on TCA and Glyoxylate Cycles.	111
Table C.4 Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on <i>S. cerevisiae</i> and <i>P. pastoris</i> promoters that regulate expression of enzymes that act on Ethanol Utilization and Alcoholic Fermentation Pathways.	117
Table C.5 Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on <i>S. cerevisiae</i> and <i>P. pastoris</i> promoters that regulate expression of enzymes that act on Glycerol Utilization Pathway.	120

Table C.6 Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on <i>S. cerevisiae</i> and <i>P. pastoris</i> promoters that regulate expression of enzymes that act on Anaplerotic Reactions.	121
---	-----

Table D.1 Changes of MCC Scores of the models as number of features is decreased	124
--	-----



LIST OF FIGURES

FIGURES

Figure 2.1	A graphical representation of working principle of TFs.	6
Figure 2.2	Transcriptional Network of formed by all TFs <i>S. cerevisiae</i> [7] .	8
Figure 2.3	A Metabolic Model for the Central Carbon Metabolism of <i>S. cerevisiae</i> [8]	9
Figure 3.1	Sequence motif, PWM [9] and calculated PSSM for Abf2 TF of <i>S. cerevisiae</i>	13
Figure 3.2	Graphical Comparison of Single Decision Tree and Random Forest Algorithms	18
Figure 3.3	Graphical Comparison of a Simple Neural Network and a Deep Neural Network.	24
Figure 4.1	A flowchart representation of Footprinting Algorithm	38
Figure 4.2	Graphical showcase of Cat8 hits on P_{ADH2} promoter.	43
Figure 4.3	(a) A segment from pairwise alignment of P_{ADH2} (top) and $P_{PAS_chr2-1_0472}$ (bottom) promoters. Msn2 TFBS annotated on P_{ADH2} , and the region its aligned against is highlighted in yellow. (b) Alignment of highlighted regions. Final Annotated sequence on <i>P. pastoris</i> is highlighted in green.	44

Figure 4.4	Comparison of <i>S. cerevisiae</i> and <i>P. pastoris</i> sequence Abf2 TFBS motifs (Generated by WebLogo [5]) and their PWMs (<i>S. cerevisiae</i> PWM was accessed from TransFac [6])	45
Figure 5.1	A Flowchart of Training and Scanning Algorithms	53
Figure 5.2	Dependency of the Model MCCs to Thresholds	57
Figure 5.3	Number of Best Models Benefited from Features elimination	60
Figure 5.4	Impact of Eliminating Features on Superior Models	61
Figure 5.5	Dependency of the Model MCCs to Thresholds when Feature Pool size is reduced to four	62
Figure 5.6	Number of superior models for specific TFs with respect to Feature set size	62
Figure 5.7	Dependency of the Model MCCs to Thresholds when Feature Pool is varied to include best performing features	63
Figure 5.8	PCA Projection of 4-mer aminoacid vectors plotted with TensorFlow Projector tool	64
Figure 5.9	PCA Projection of protein vectors plotted with TensorFlow Projector tool	65
Figure 5.10	Change of Validation Loss during training 3-depth ELU model over 1000 Epochs	66

LIST OF ALGORITHMS

ALGORITHMS

Algorithm 1	Needleman-Wunsch Algorithm [10]	16
Algorithm 2	Gradient Descent Algorithm [11, 12]	26
Algorithm 3	ADAM Algorithm [13]	28
Algorithm 4	ADAMax Algorithm [13]	29
Algorithm 5	Phylogenetic Footprinting Algorithm	39
Algorithm 6	PWM Motif Scanning	40
Algorithm 7	ScanAlgo Algorithm	41
Algorithm 8	Greedy Algorithm used to model TF interactions	52

LIST OF ABBREVIATIONS

ADAM	Adaptive Moment Estimation
ADH2	Alcohol Dehydrogenase 2
BLAST	Basic Local Alignment Search Tool
ChIP	Chromatin Immunoprecipitation
CNN	Convolutional Neural Network
DWM	Dinucleotide Weight Matrix
ELU	Exponential Linear Unit
FN	False Negative
FP	False Positive
PseKNC	Pseudo-k-tuple Nucleotide Composition
PseEIIP	Electron-Ion Interaction Pseudopotential
HMM	Hidden Markov Model
MCC	Mathews' Correlation Coefficient
MLP	Multilayer Perceptron
PBM	Protein Binding Microarray
PSSM	Position-Specific Scoring Matrix
PWM	Position Weight Marix
ReLU	Rectified Linear Unit
SGD	Stochastic Gradient Descent
TF	Transcription Factor
TFBS	Transcription Factor Binding Site
TN	True Negative
TP	True Positive
XGBoost	Extreme Gradient Boosting

CHAPTER 1

INTRODUCTION

Yeasts are a popular topic of microbial research, as they have been used successfully to express a multitude of proteins, and they have many advantages such as short doubling time, a readily manipulated genome, improved protein folding, and most post-translational modifications [14,15]. *Saccharomyces cerevisiae* was the first yeast to be used for recombinant protein production [16]. Despite this, over-time *Pichia pastoris* became the premier choice for yeast expression systems, due to its tightly regulated, efficient promoters and its strong tendency for respiratory growth as opposed to fermentative growth [15, 17]. Another appealing feature of *P. pastoris* is its ability of reaching high cell densities under appropriate culture conditions, being able to reach 120 g/l dry cell weight density using inexpensive medium [14, 18].

Cells live in complex environments, and fluctuations on physical parameters (e.g. pH, temperature, or osmotic pressure) or internal state of the cell (e.g. concentration of key metabolites or DNA damage) may occur. To represent these complex environmental states, cells use special proteins called “*Transcription factors*” (TFs) [19]. TFs respond to these conditions by switching between their active and inactive states, and each active TF can bind to DNA to regulate the rate of expression of its target gene [19]. Activation of signalling pathway(s) in the cell that alters crosstalk involves activation of TF(s) coordinated with specific *cis*-acting DNA sequences (cADSs) in the upstream regions of the promoter(s) [20].

Metabolic engineering is a discipline that aims to increase the productivity of a cell, by optimizing the metabolism of the organism and purposeful modification of its gene regulatory networks [21]. In traditional metabolic engineering, foreign enzymes are expressed at constant levels from inducible or constitutive promoters. However, this

way of engineering corresponds to an open-loop strategy, and the engineered system cannot respond to fluctuations that may occur in the cellular environment [22]. Knowledge of regulatory regions such as *TF binding sites* (TFBSs) is imperative for metabolic engineering as it allows manipulations on regulatory system of the cell. As such, identifying TFBSs is compelling, therefore popular subject of study. As experimental identification of all binding sites of a TF for every cell type and operation condition remains infeasible, using computational tools to predict experimentally unidentified TFBSs is gaining momentum. Among these tools, the most widely used are position weight matrix, dinucleotide weight matrix, TF flexible model, pairwise interaction model and machine learning models [23].

Engineering of promoter architectures with de novo synthetic biology tools for tailoring de novo production strategies in heterologous protein production has conferred breakthrough success in yeast [24, 25]. The design of synthetic promoter architectures [26, 27] hinges on genomic and functional annotation. *Saccharomyces cerevisiae* is the first model yeast. Its databases host genomic and functional annotation information [28, 29]. The widely used, creatively engineered but relatively less studied yeast *Pichia pastoris* (syn. *Komagataella phaffii*) has advantages in recombinant protein (r-protein) production. The advantages are high production capacity, stress tolerance, robustness, genetic accessibility, simple nutritional requirements, and reaching high cell densities [15, 18, 30–32]. In contrast to *S. cerevisiae*, *P. pastoris* lacks extensive functional annotation studies. However, it is ideally placed taxonomically to make annotation propagation from *S. cerevisiae* highly informative [33].

In this Thesis, three computational methods, Phylogenetic Footprinting, Machine Learning, and Machine Learning with Word2Vec, are presented to predict the TFBSs in yeast promoters. As *S. cerevisiae* has the most significant number of transcription factor binding specificity datasets, its datasets were used to prepare the models. For Phylogenetic Footprinting, *S. cerevisiae* promoters has been scanned for TFBSs using datasets available for *S. cerevisiae* TFs. The results were compared against homologous *P. pastoris* promoters, and the conserved TFBSs are determined. Using conserved TFBSs, PWMs of *P. pastoris* TFs has been predicted. For Machine Learning methods, five features were employed to encode the data as an input vector to train the models to recognize 8-mer TFBSs: *k*-mer dinucleotide frequency, *k*-spaced

nucleotide pair frequency, and nucleotide chemical property pseudo nucleotide composition, and electron-ion interaction pseudo-potentials of trinucleotide, similar to Wang et al. [34]. For regular Machine Learning, for all TFs of *S. cerevisiae* an individual model was developed. For Machine Learning with Word2Vec [35], TFs have been represented as vectors using Word2Vec and a single model that can predict the TF-TFBS interaction was developed. For both Machine Learning models, various Machine Learning algorithms were used and the best-case algorithms were compiled into a library, which was then used to design a scanning algorithm that can determine the putative TFBSs in yeast promoters. Further, a module that can be utilized by the users to develop their own models is also presented.





CHAPTER 2

BIOLOGICAL BACKGROUND

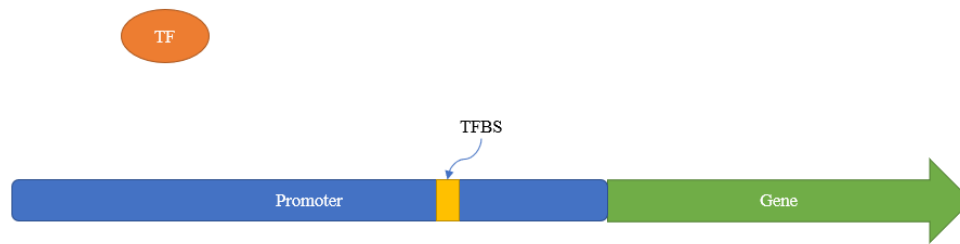
2.1 Transcription Factors

Transcription Factors (TFs) are regulatory proteins that bind to specific DNA sequences called *Transcription Factor Binding Sites* (TFBSs) to control gene expression rates [36]. RNA Polymerase, responsible for mRNA production, binds to specific sequences that are directly adjacent to the gene that is to be transcribed, which are called promoters [37]. TFBSs reside in these promoter regions of the DNA, which TFs bind to, either to help activate or to repress this transcription process that leads to increased or decreased protein synthesis, respectively [37].

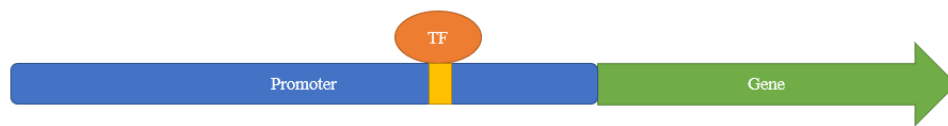
TFs can affect expression positively or negatively, depending on interaction with other elements of transcription such as mRNA or other TFs [36]. A TF can upregulate expression by forming stable transcriptional complexes with another already bound factor; or repress the expression by simply occupying a binding region, preventing a transcriptional element that is required to start transcription from binding [36].

To control gene expression in a meaningful way, there are some means exist to modulate the activity of specific TFs [36]. TFs bind to DNA when activated, which may be caused by various external signals that carry information about the environment, specific to the TF [19, 37]. A graphical demonstration of TFs general working principle is presented in Figure 2.1.

Understanding working mechanisms of TFs, determining TFBSs, and understanding their interactions with the genes is very important for disciplines such as Biochemical or Metabolic Engineering, as this information can be used to engineer cells for



(a) In inactive state, TFs do not bind to DNA.



(b) Upon activation, TF finds nearest recognized site and binds to it.

Figure 2.1: A graphical representation of working principle of TFs.

overexpressing certain genes to achieve improved production of certain biochemicals.

To identify TFBS sequences, there are a number of experimental methodologies developed, such as ChIP-seq or PBMs. *Chromatin Immunoprecipitation* (ChIP), is a method that has been proposed in 1988 by Solomon et al. [38], to study protein interactions with DNA. Working mechanism of the ChIP can be summarized in 3 stages [39]:

1. Formation of covalent cross-links between the protein of interest and DNA ,
2. Formation of covalent cross-links between the protein of interest and a specific antibody that can be used to coimmunoprecipitate,
3. Immunoprecipitation of protein-DNA pairs that were formed in stage 1, which can then be reversed to recover the DNA sequences bound by the protein.

Ren et al. [40] has proposed ChIP-chip in 2000, improving the methodology by using

a mock sample that does not have antibodies added in 2nd step. This addition allows for filtering of non-specific DNA that might be produced by the ChIP procedure due to random DNA-protein binding, which is a significant advantage [39].

One other variation of ChIP is ChIP-seq, where ChIP technology is used in conjunction with the DNA sequencing technologies, such as Solexa [41], which provides short, ideal DNA sequences for ChIP-sequencing [39]. This variation of the ChIP was found advantageous to ChIP-chip, as it required much less hand-on processing, was cheaper and required less input and replicates to process [39]. Robertson et al. [42] showed that ChIP-seq while has these advantages, does provide highly similar results compared to ChIP-chip for Stat1 binding sites for human HeLa3 cells.

While ChIP, ChIP-chip, and ChIP-seq are powerful *in-vivo* tools for TFBS determination, they are low throughput, as they check for only nucleotide subsequences within a given DNA sequence. *Protein Binding Microarrays* (PBMs), on the other hand, is a tool that can be used to determine a TFs *in-vitro* interactions to all possible 4^k k-mer sequences, where k stands for the length of the sequence [43]. Main limitations of the PBMs is the sequence length, since as k increases number of sequences that need to be checked increases exponentially, and possibility of *in-vitro* results disagreeing with *in-vivo* results [43]. PBMs construct universal microarrays where signalling frequencies can be used to measure and score the binding affinities of TFs to k-mer sequences simultaneously, which is the biggest advantage of PBMs [43].

There are a number of public databases available to reach TFBS data for both types of methodologies, such as Jaspar [9], Transfac [6] or Yeastract [44] (only *S. cerevisiae*) for ChIP-chip data, and UniProbe [45] for PBM data, which were utilized in the scope of this study.

2.2 Cell Lines

2.2.1 *Saccharomyces cerevisiae*

S. cerevisiae, also known as the baker's yeast, is the traditional biotechnological organism, and the first eukaryote ever to receive a complete genome sequencing [15,46].

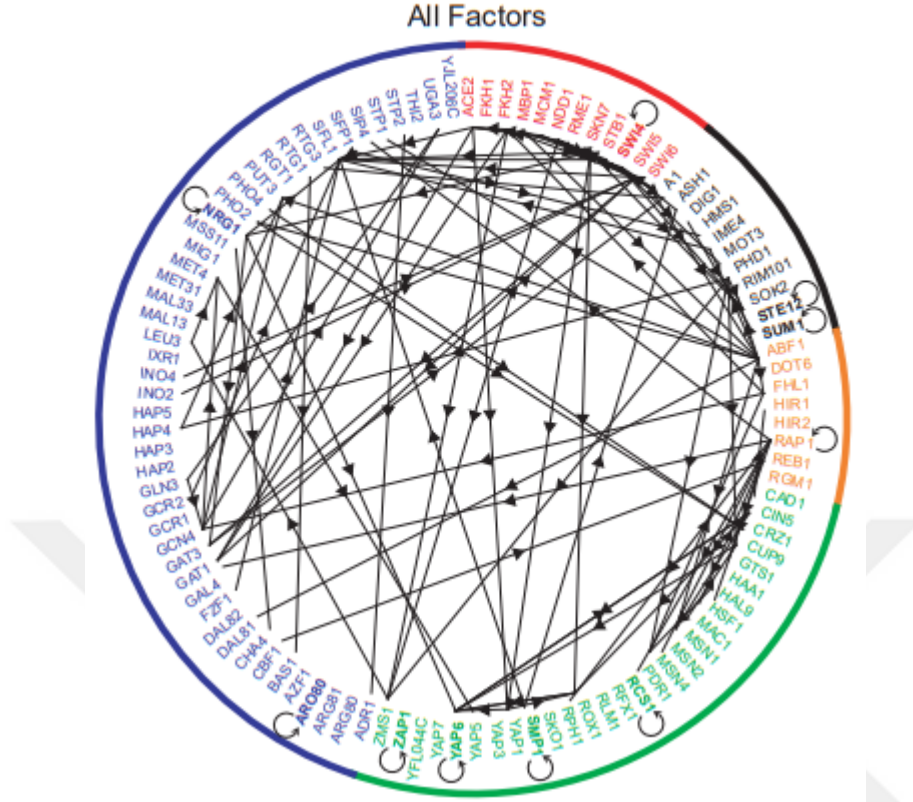


Figure 2.2: Transcriptional Network of formed by all TFs *S. cerevisiae* [7]

S. cerevisiae is a Crabtree-positive yeast, which means it prefers to ferment even in the presence of sufficient oxygen and glucose [47]. *S. cerevisiae* has been the model organism in understanding biological pathways and regulatory networks for nearly 30 years, and numerous studies with high-throughput experiments has been done on understand its transcriptional network [48,49]. The transcriptional network of *S. cerevisiae* is shown in Figure 2.2 [7].

Availability of extensive and public knowledge for *S. cerevisiae* makes it an incredible reference organism when used for modelling purposes. Important databases exist such as KEGG [50], Yeasttract [44], JASPAR [9], TransFac [6], UniProbe [45], UniProt [51], and NCBI [52] which can be used to access information about *S. cerevisiae*'s genome, TFBSs, protein sequences or enzyme duties.

S. cerevisiae, as an expression system has most of the advantages that are attributed

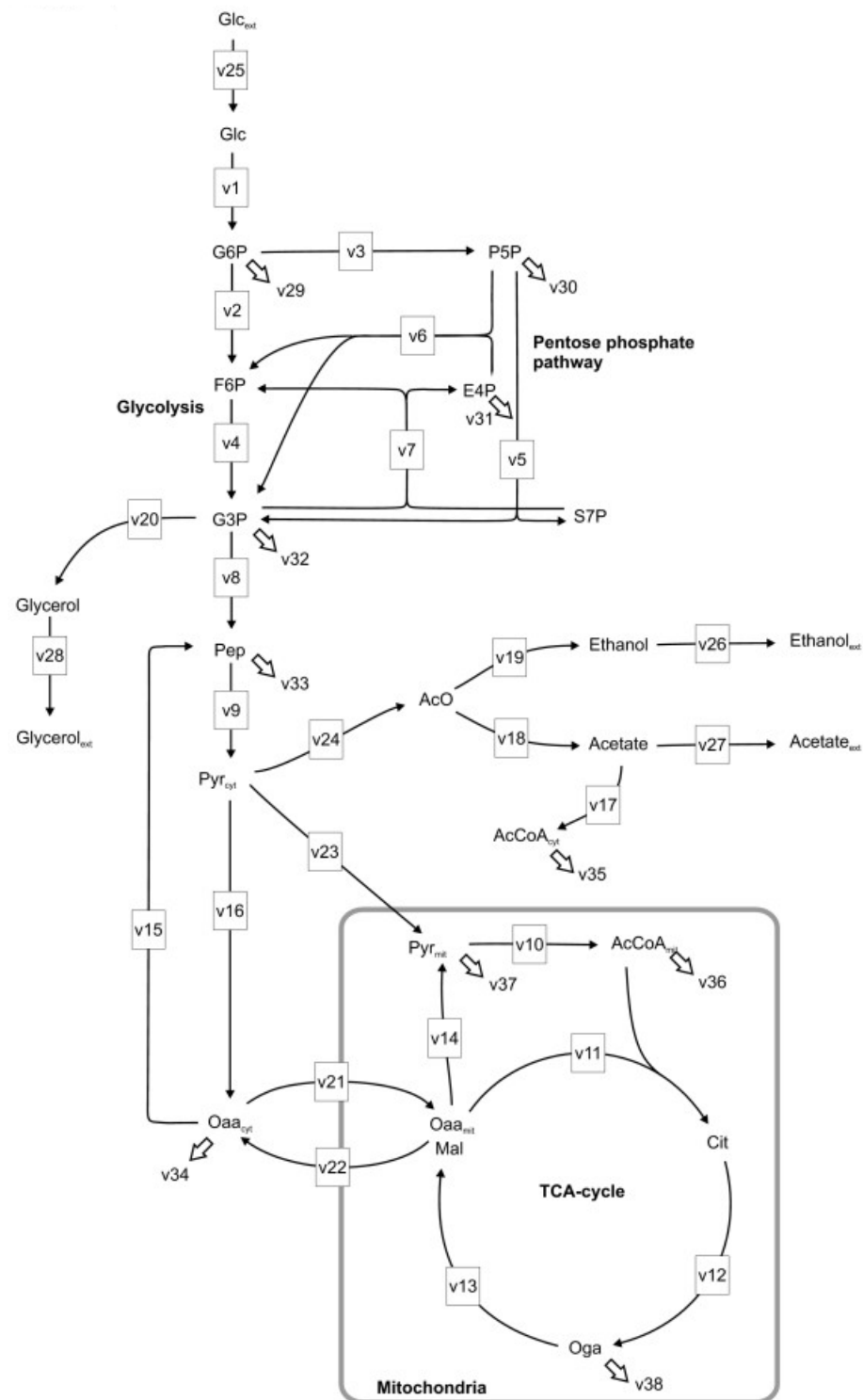


Figure 2.3: A Metabolic Model for the Central Carbon Metabolism of *S. cerevisiae* [8]

to yeasts, such as the ability to grow in cheap media or ability to do post-translational modifications, while having some further benefits such as having high tolerance for environmental conditions or the highest glycosylation capacity over all yeasts [15]. These benefits cause it to be recognized as a generally regarded as safe (GRAS) type of expression system [15]. The central carbon metabolism reactions of *S. cerevisiae* is presented Figure 2.3.

2.2.2 *Pichia pastoris*

As an expression system *P. pastoris* holds significant advantages over *S. cerevisiae*, being a Crabtree negative yeast, which causes it to reach much higher cell concentrations, as products of fermentation such as acetic acid or ethanol do not accumulate in the environment [53]. Another advantage that *P. pastoris* has over *S. cerevisiae* is its tendency to secrete its proteins, even those that has high molecular weight, instead of keeping them in the periplasm [15].

P. pastoris's whole genome sequence was not available until 2009 [54], while *S. cerevisiae* was sequenced in 1996 [46]. Thus, despite *P. pastoris* having significant advantages compared to *S. cerevisiae* the difference in knowledge between two species is vast. Thus, while studying *P. pastoris*, using *S. cerevisiae* as a model organism is common [24, 25].

CHAPTER 3

THEORETICAL BACKGROUND

3.1 Phylogenetic Footprinting

Phylogenetic footprinting is a method for finding putative cADSs by comparing the upstream regulatory region of the gene with its orthologues from different species through sequence alignment. It is based on the hypothesis that the functional elements in the non-coding DNA regions are conserved as their evolution is under selective pressure. Thereby, they evolve slower than their non-functional surrounding [55]. Through the curation pipelines to be established, the flow of the extensive cellular knowledge that belongs to *S. cerevisiae* enables predictions to fill the functional annotation gap in other yeasts in the short term.

3.1.1 Position Weight Matrices and Motif Searching

Most TFs recognize a set of short DNA sequences between 5-20 bp. The recognized DNA sequences for a specific TF are bound to have common specific features. *Consensus sequence*, is a concept that has been widely used to represent the specificity of TFs [56]. Defining a consensus sequence for a set of sequences tends to be arbitrary, as there is a trade-off between allowed mismatches and ambiguity of the consensus [56]. When trying to predict new TFBSs, if a consensus sequence is too strict, an important chunk of functional TFBSs will be missed; similarly, if a consensus sequence is too ambiguous, the number of false-positive results will be large [56]. As such, while it is easy to represent the set of recognized sequences of a TF using consensus sequences, they are not as useful when trying to predict new ones [56].

Representing known TFBSs for a TF utilizing a *Position Weight Matrix* (PWM) allows for a quantitative description of consensus sequence [56,57]. By assuming each position of the TFBS sequence contributes independently to the binding energy, these weight matrices can be used to create *Positions-Specific Scoring Matrices* (PSSMs), which assign scoring values for each position within the TF motif that judges the contribution made by a given nucleotide in that position [56,57]. Conversion between a PWM to a PSSM can be done by dividing the observed nucleotide probabilities to expected background probabilities and converting it to log-scale as shown in Equation 3.1 [56–58]:

$$W(b, i) = \log_2 \frac{p(b, i)}{p(b)} \quad (3.1)$$

where $W(b, i)$ is the score assigned to nucleotide b in position i , and $p(b)$ is the background probability of observing nucleotide b .

As known TFBSs may not represent the precise nature of the consensus motif, a sampling correction must be done utilizing by adding pseudocounts, as shown in Equation 3.2 [57]:

$$p(b, i) = \frac{f_{b,i} + s(b)}{N + \sum s(b)} \quad (3.2)$$

where i is the position, $p(b, i)$ is the corrected probability of observing nucleotide b in position i , $f_{b,i}$ is the number of nucleotide b has been observed in position i , N is the number of experiments, and $s(b)$ is the correction factor, also known as the pseudocount.

PWMs also allow for easy visualizations for the consensus sequences, by the use of "*sequence logos*", by stacking letters on top of each other while their height is proportional to their observed frequency [59]. Total heights in columns in the sequence logos gives the information content of the sequence at that position [59], which can also be calculated with the help of Equation 3.3 [56,60]:

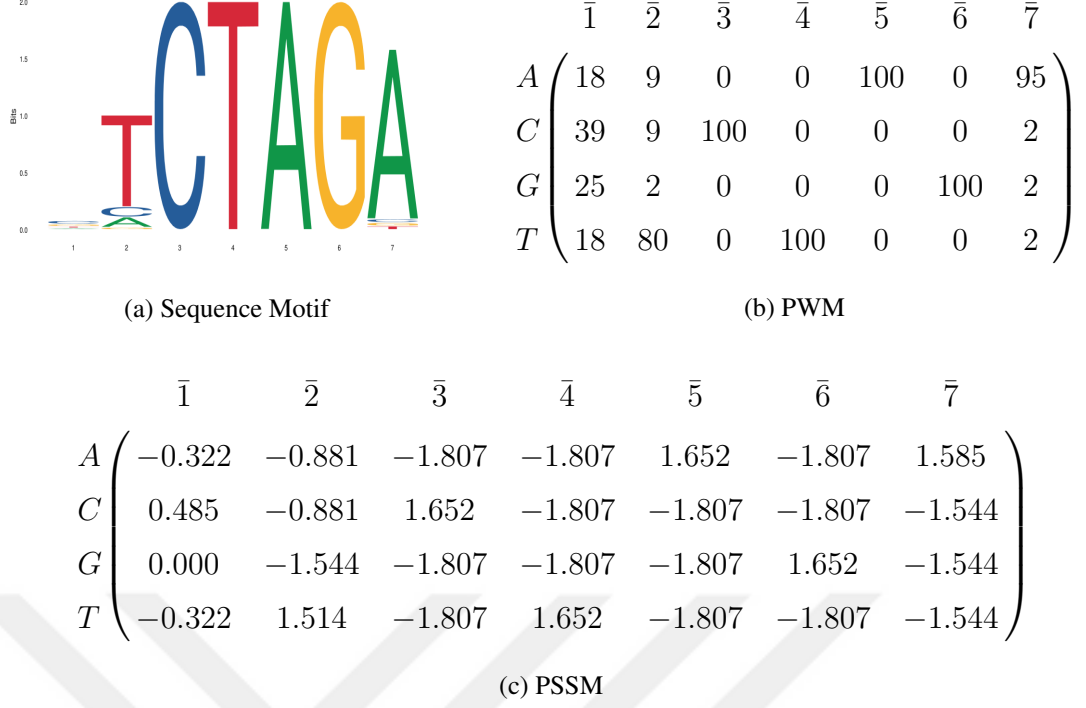


Figure 3.1: Sequence motif, PWM [9] and calculated PSSM for Abf2 TF of *S. cerevisiae*

$$I_i = 2 + \sum_{b=A}^T f_{b,i} \log_2 f_{b,i} \quad (3.3)$$

where i refers to the position, I_i is information content at a given position, b is for base, and $f_{b,i}$ is the observed frequency of a given base at a given position.

An example sequence motif, PWM and PSSM is provided for Abf2 TF of *S. cerevisiae* in Figure 3.1. Uniform background probabilities were assumed while calculating the PSSM.

After knowing the PSSM for a TF, information content of any sequence A can be determined by taking the dot product of the sequence against the PSSM [56], which can be used as a scoring mechanism when trying to find new TFBSs. Equation 3.4 showcases the formula used to calculate the information content of a sequence [56, 57]:

$$I_A = \sum_{i=1}^{l_A} W_{A_i,i} \quad (3.4)$$

where I_A is information content of A , i is the position, l_S is the length of A , and $W_{S_i,i}$ is the PSSM score of base of A at i th position in i th column.

Calculating a relative information content is also useful when using PWMs, formula of which is given in Equation 3.5, which allows to judge a sequence based on how close it is to having maximum information content [9]:

$$S = \frac{\sum_{i=0}^{i=L_A} W(b_{A,i}, i) - \sum_{i=0}^{i=L_A} W_{min}(i)}{\sum_{i=0}^{i=L_A} W_{max}(i) - \sum_{i=0}^{i=L_A} W_{min}(i)} \quad (3.5)$$

where S is the relative score, L_A is the length of sequence A , $b_{A,i}$ is the nucleotide observed in position i , $W_{min}(i)$ is the minimum score observed in PSSM in position i and $W_{max}(i)$ is the maximum score observed in PSSM in position i .

The PWMs still carry some amount of ambiguity when searching for new sites, as how high information content needs to be accepted as a TFBS still is up to personal interpretation. Since, in natural genomic context, exon ≥ 2 sequences do not contain any regulatory regions [61], a good rule of thumb is to set thresholds such that negligible amount of matches will be found when scanning these sequences [62]. Another good rule of thumb is to define core regions, by determining nucleotides that are undisputed in their respective at their respective positions, and eliminate any matches that does not share these core regions, no matter how high their scores are [62].

3.1.2 Introduction to Phylogenetic Footprinting

Although PWMs can and do predict most of the known functional TFBSs, they are not very trustworthy upon analyzing large scale uncharacterized sequences, as they lack the mechanism to filter out the false-positive predictions, as a result of the nature of binding sites [63]. As such, combining them with other tools that can filter out the false-positive predictions is vital. One such approach is to draw comparison of orthologous sequences from multiple species, termed as "*Phylogenetic footprint-*

ing" [64]. Underlying idea behind phylogenetic footprinting is that selective pressure causes more relevant set of sequences will evolve in a slower rate, compared to non-functional surrounding sequence [65,66]. Selection of species that are to be compared are essential for accurate results in phylogenetic footprinting, as too closely related species would result in high number of false positives, while too distant species would be impossible to find conserved motifs [67]. The sequence similarity in DNA regions, that are not subject to selective pressure, between two species that have diverged 300 million years ago have been estimated to be about 30%, approximately same as two unrelated species [67]. Thus, this can be a strong indication of functionality for conserved DNA elements between two such species [67]. The divergence between *S. cerevisiae* and *P. pastoris* dates back 250 million years [68], making them plausible phylogenetic footprinting candidates.

Phylogenetic footprinting is to be a contrast to a what used to be a much more common approach of considering a group of related genes for a single species and creating multi-sequence alignments [69]. The multi-gene approach has an inherent limitation, which is that they will only find regulatory elements that are common to a number of genes. While multi-species approach is capable of identifying regulatory elements that are specific to a single gene, as long as they are conserved across several species [69], which makes it more advantageous.

3.1.3 Determination of Homologous Sequences

Phylogenetic footprinting requires homology knowledge between sequences to implement. Basic Local Alignment Search Tool (BLAST), is a rapid, robust and a simple tool that allows the detection of biologically significant sequence similarities [70]. BLAST algorithm starts by taking an input query, a database of sequences that will be searched against, a scoring matrix and a threshold (T) [70]. BLAST quickly identifies the sequences that have lower chances of exceeding T by checking whether or not it shares a word of length w that can pair with the query, and minimizes time spent with these sequences [70]. After a shrinking the range of search, it produces alignment scores against only a handful sequence pair, employing the Needleman-Wunsch algorithm (Algorithm 1) [10], and an expectancy value, formula of which is provided

Input 1: Two sequences, A and B, to be aligned

Input 2: A scoring matrix

Step 1: Start by creating a $(m+1) \times (n+1)$ matrix H where m and n are the lengths of the sequences that are to be aligned.

Step 2: Set $H_{0,0}$ to 0; and $H_{i,0}$ and $H_{0,j}$ to $i * W$ and $j * W$ for all values of i and j , respectively, where W is the gap penalty.

Step 3: For all values of i and j , calculate:

$$H(i, j) = \max \left\{ \begin{array}{l} H(i-1, j) + W \\ H(i, j-1) + W \\ H(i-1, j-1) + S_{A_i, B_j} \end{array} \right\}$$

where, S_{A_i, B_j} is the reward/penalty for association of characters A_i and B_j , read from the scoring matrix.

Output: $H_{m+1, n+1}$ as the alignment score.

Algorithm 1: Needleman-Wunsch Algorithm [10]

in Equation 3.6 [70]:

$$e = Kmn * \exp(-\lambda * S) \quad (3.6)$$

where, S is the alignment score, m and n are the sequence lengths, K and λ are parameters tuned based on the database size and distribution of similarity scores of unrelated sequences [71], and e is the number of expected sequences that would achieve score S or higher, for the given sequence lengths and database size.

3.1.4 TFBS Conservation Criterion

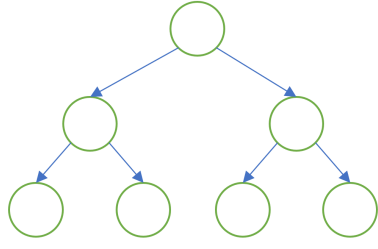
When applying phylogenetic footprinting between two homologous sequences, there might be TFBSs that align to significantly similar regions, but with point mutations that changed them during the evolutionary cycle. While not all TFs have the same specificity when selecting binding sites, a binding probability that is higher than 2/3 (66%) can indicate a strong affinity against that site [72].

3.2 Machine Learning

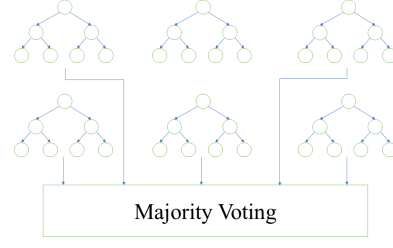
3.2.1 Transition to Machine Learning

As explored in Section 3.1.1, PWMs work under the assumption of each position of a sequence contributes to the binding energy independently, which does not agree to experimental results [73–75]. *Dinucleotide Weight Matrix* (DWM) is the first method to be introduced to deal with this issue [76]. DWMs showcase the probabilities of observing any nucleotide pair at any position which allows them to perform significantly better than PWMs for most TFs [76]. While this gives some hints about the contribution of the nucleotide groupings, it still does not explore neither the contribution made by longer nucleotide groupings, nor the contributions of nucleotide pairings that are apart from each other. *Transcription Factor Flexible Model* (TFFM) was built on the concept of DWMs, which is a Hidden Markov Model (HMM) based prediction model [77]. While TFFMs did not address the issue of contribution of longer nucleotide groupings, the HMM-based framework was flexible, supported contribution of dinucleotide pairings and variable lengths of TFBS motifs, and did perform significantly better than its predecessors [77]. The contributions of nucleotide pairings were first investigated by *Pairwise Interaction Model* (PIM), a model that uses the brute-force approach to enumerate and compute the binding energy of all possible k-mers, where k is the sequence length of the TFBS of concern [78]. While this technique allowed them to bypass several simplifying assumptions, it demands a much heavier computational-power [78].

As high-throughput techniques for measuring *in vitro* protein-DNA binding, such as PBMs, kept advancing, more robust options with less simplifying assumptions were required, as biological data are very susceptible to noise and bias, which has led to an increase using Machine Learning and Deep Learning algorithms for TFBS predictions [23]. Predicting TFBSs involves taking an input k-mer sequence and mapping it to a 1 (is a TFBS) or a 0 (not a TFBS). Machine Learning procedures where the machine takes a set of example input-output pairs, and learns a function that maps them into each other, such as the case of predicting TFBSs, is called supervised learning [79]. The most common form of Machine Learning (deep or not) is supervised learning



(a) Single Decision Tree Algorithm



(b) Random Forest Algorithm

Figure 3.2: Graphical Comparison of Single Decision Tree and Random Forest Algorithms

[80].

3.2.2 Random Forest

Random Forest is a classifier that consists of a set of tree-structured classifiers, each voting for the answer [81]. Random Forests follow the strong law of large numbers, which states that as the number of independent and identical experiments increase, there is a probability of 100% that the mean of the results will converge to a constant value, as formulated by Equation 3.7 [81, 82]:

$$P \left[\lim_{n \rightarrow \infty} |\bar{S}_n - \mu_x| \leq \varepsilon \right] = 1 \quad (3.7)$$

where, P is the probability, $\bar{S}_n$ is the mean of results for n experiments, μ_x is the true probability of occurrence of x , and ε is an infinitesimal margin.

Strong law of large numbers dictates that as number of decision trees increase, a limiting value on the generalization error, the measure of how accurately unseen data can be predicted by the algorithm [83], is reached, which means they are prone to overfitting much-less than most single decision tree algorithms [81]. A graphical comparison of a single decision tree algorithm and a Random Forest algorithm is given in Figure 3.2.

Random Forests uses an out-of-bag error estimation model, where a sample is drawn,

with replacement, from the dataset and each tree is grown with a bag of training set using random feature selection to create bagged predictors [81]. After the bagged predictors is grown, for each data/label pair in the dataset, only the votes of predictors that did not had access to that pair gets aggregated to estimate the generalization error [81]. Out-of-bag error estimation was shown to be as accurate as a train-test splitting method, which eliminates the need to set aside a test set [81].

Random Forest algorithms allow for additional random elements can be introduced, such as column subsampling, which is a method of limiting the splitting candidates to a small, randomly selected subset of features for each split in the trees [81,84]. While growing classification and regression trees, the split points are found by maximizing the decrease of the impurity function [84–86]. For classification trees, the impurity is usually measured by Gini Impurity, shown in Equation 3.8 [84, 86]:

$$\hat{\Gamma}(t) = \sum_{j=1}^J \hat{\phi}_j(t)(1 - \hat{\phi}_j(t)) \quad (3.8)$$

where, $\hat{\Gamma}(t)$ is the impurity of node t , and $\hat{\phi}_j(t)$ is the class frequency for class j for node t .

Random Forests has been used in recent years to predict TFBSs [87–89], for their advantages being:

- consideration of many decision trees, which reduces the chance of overfitting and misclassification [88],
- ability to achieve high model complexity and is able to process large genetic data sets obtained with high-throughput methods [89],
- being faster and simpler to work with compared to other Machine Learning methods,
- eliminating the need for a test data [81],

which allow them to be useful tools when working with genetic data. However, usage of many decision trees instead of just one results in a much more complex model, resulting in a harder environment for interpreting feature importance.

3.2.3 Extreme Gradient Boosting

Extreme Gradient Boosting (XGBoost) is a scalable Machine Learning system that is based on gradient boosting method [90]. Just like Random Forests, boosting is a method that aims to combine weak classifiers to form a strong ensemble [91]. While a single classifier system aims to find a function $F(x)$ that minimizes a specified objective function $\Psi(y, F(x))$ over a training set, boosting approaches the problem on an additive manner (Equations 3.9 and 3.10) [92]:

$$F(x) = \arg \min_{F(x)} \Psi(y, F(x)) \quad (3.9)$$

$$F_M(x) = \sum_{m=0}^M \beta_m h_m(x) \quad (3.10)$$

where, $h_m(x)$ is a weak classifier that is a simple function of x (e.g. a *Decision Tree*) that has the index m , β_m is the weight given to the m th weak classifier, $F(x)$ is the resulting model for a single classifier system, and $F_M(x)$ is the resulting model for a boosting system after M iterations [92].

While regular tree generation is an additive process and traditional methods of optimization cannot be used, boosting is an iterative procedure, where the objective is to finding the next weight, weak classifier pair that minimizes the objective function, which is the sum of loss functions f for all N trees as shown in Equations 3.11 and 3.12 [90, 92, 93]:

$$(\beta_m, h_m(x)) = \arg \min_{(\beta_m, h_m(x))} \Psi(\beta_m, h_m(x)) \quad (3.11)$$

$$\Psi(\beta_m, h_m(x)) = \sum_{m=1}^N l(y, (F_{m-1}(x) + \beta_m h_m(x))) \quad (3.12)$$

Gradient boosting [93] uses a two-step procedure to find an approximate solution pair $h_m(x)$, β_m (Equation 3.12) [92, 93]. First, the weak classifier $h_m(x)$ that fits to pseudo-residuals the best is determined using the least square method (Equations

3.13 and 3.14), which is then assigned an optimal weight β_m , that minimizes the loss function l (Equation 3.15) [92, 93]:

$$\tilde{y}_{i,m} = - \left[\frac{\delta(l(y_i, F(x_i)))}{\delta(F(x_i))} \right] \quad (3.13)$$

$$h_m(x) = \arg \min_{h_m(x), \rho} \sum_{i=1}^N [\tilde{y}_{i,m} - \rho h_m(x)]^2 \quad (3.14)$$

$$\beta_m = \arg \min_{(\beta_m)} \sum_{m=1}^N l(y, (F_{m-1}(x) + \beta_m h_m(x))) \quad (3.15)$$

where, $\tilde{y}_{i,m}$ is for pseudo-residuals.

In Gradient Tree Boosting, the weak classifier $h_m(x)$ is a tree with L terminal nodes, that divides the data to L categories which are all assigned with a separate constant $\bar{y}_{l,m}$, based on the mean of pseudo-residuals ($\tilde{y}_{i,m}$) of the data that fall into those categories [92, 93]. Thus, the weighing factor is eliminated, and Equations 3.11 and 3.12 becomes:

$$h_m(x) = \arg \min_{h_m(x)} \Psi(h_m(x)) \quad (3.16)$$

$$\Psi(h_m(x)) = \sum_{m=1}^N l(y, (F_{m-1}(x) + h_m(x))) \quad (3.17)$$

where $h_{m,l}(x)$ is a tree of L terminal nodes generated in m th iteration [92, 93].

XGBoost is a gradient tree boosting system that introduces several novelties [90]. Firstly, a penalty for the model complexity is introduced (Equation 3.18), which pushes the algorithm to select simpler functions [90]. With the introduction of this term, Equation 3.17 takes the form of 3.19 [90]:

$$\Omega(h_{m,L}(x)) = \gamma L + \frac{1}{2} \lambda \sum_{l=1}^L \bar{y}_{l,m}^2 \quad (3.18)$$

$$\Psi(h_{m,L}(x)) = \sum_{m=1}^N l(y, (F_{m-1}(x) + h_{m,L}(x))) + \Omega(h_{m,L}(x)) \quad (3.19)$$

where Ω is the penalty for model complexity and γ and λ are user selected constants.

XGBoost uses a second-order Taylor expansion to approximate first term in Equation 3.19, which results in Equation 3.22 [90]:

$$g_i = - \left[\frac{\delta(l(y_i, F(x_i)))}{\delta(F(x_i))} \right] \quad (3.20)$$

$$h_i = - \left[\frac{\delta^2(l(y_i, F(x_i)))}{\delta(F(x_i))^2} \right] \quad (3.21)$$

$$\Psi(h_{m,L}(x)) = \sum_{m=1}^N \left[l(y, F_{m-1}(x)) + g_i h_{m,L}(x) + \frac{1}{2} h_i h_{m,L}(x)^2 \right] + \Omega(h_{m,L}(x)) \quad (3.22)$$

The first term in the brackets is independent of the current iteration, hence it can be dropped safely to get Equation 3.23 [90]:

$$\Psi(h_{m,L}(x)) = \sum_{m=1}^N \left[g_i h_{m,L}(x) + \frac{1}{2} h_i h_{m,L}(x)^2 \right] + \Omega(h_{m,L}(x)) \quad (3.23)$$

If the instance set of leaf l is defined as I_l (Equation 3.24), Equation 3.23 can be rewritten as Equation 3.25 [90]:

$$I_l = \{i | h(x_i) = l\} \quad (3.24)$$

$$\begin{aligned} \Psi(h_{m,L}(x)) &= \sum_{m=1}^N \left[g_i h_{m,L}(x) + \frac{1}{2} h_i h_{m,L}(x)^2 \right] + \gamma L + \frac{1}{2} \lambda \sum_{l=1}^L \bar{y}_{l,m}^2 \\ &= \sum_{m=1}^N \left[\left(\sum_{i \in I_l} g_i \right) \bar{y}_{l,m} + \frac{1}{2} \left(\sum_{i \in I_l} h_i + \lambda \right) \bar{y}_{l,m}^2 \right] + \gamma L \end{aligned} \quad (3.25)$$

For a fixed tree structure $h_L(x_i)$, one can compute the optimal weight $\bar{y}_{l,m}^*$ of a leaf which minimizes the function in brackets by taking derivative with respect to leaf weight $\bar{y}_{l,m}$ which will result in Equation 3.26 [90]:

$$\bar{y}_{l,m}^* = -\frac{\sum_{i \in I_l} g_i}{\sum_{i \in I_l} h_i + \lambda} \quad (3.26)$$

which then can be plugged to the loss function to achieve Equation 3.27 [90]:

$$\Psi(h_{m,L}(x)) = -\frac{1}{2} \sum_{m=1}^N \frac{(\sum_{i \in I_l} g_i)^2}{\sum_{i \in I_l} h_i + \lambda} + \gamma L \quad (3.27)$$

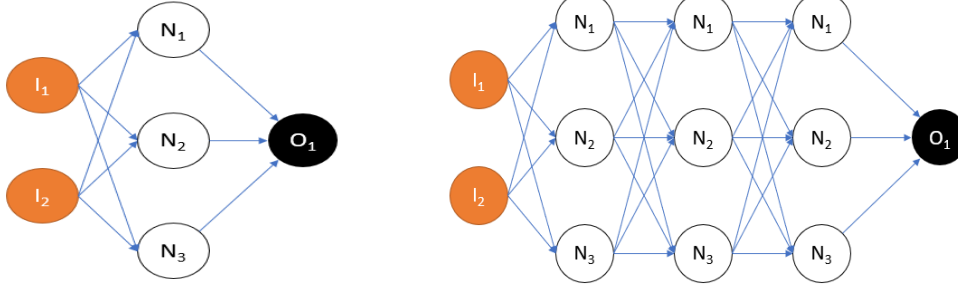
Equation 3.27 allows one to measure the quality of any decision tree, which can then be used to grow trees using a greedy algorithm [90]. However, since it is not possible to evaluate every possible tree, starting with a single tree and adding branches to it by selecting splitting candidates from finite number of features is used instead [90]. This is done so by modifying the Equation 3.27 such that the loss prior to split is compared to loss after the split, and a gain function is written accordingly (Equation 3.28) [90]:

$$Gain = \frac{1}{2} \left[\frac{(\sum_{i \in I_{left}} g_i)^2}{\sum_{i \in I_{left}} h_i + \lambda} + \frac{(\sum_{i \in I_{right}} g_i)^2}{\sum_{i \in I_{right}} h_i + \lambda} - \frac{(\sum_{i \in I} g_i)^2}{\sum_{i \in I} h_i + \lambda} \right] - \gamma \quad (3.28)$$

where I_{left} and I_{right} are the instances that are split to left and right child nodes after the split, respectively.

When growing trees, XGBoost aims to determine the best splitting candidate that maximizes the gain function [90]. XGBoost also utilizes column subsampling and shrinkage, the method of scaling each generated tree with a learning rate factor, to combat overfitting [90].

XGBoost was proven to be an excellent tool that has been used with success in various fields, including classifying genetic data, achieving a 79.72% accuracy when used to classify 8-mer DNA sequences into TF families [34].



(a) A single layer (Simple) Neural Network

(b) A multiple layer (Deep) Neural Network

Figure 3.3: Graphical Comparison of a Simple Neural Network and a Deep Neural Network.

3.2.4 Deep Learning

Representation learning is a set of methods, which the machine is fed with raw data and extract the features required for classification automatically [80]. *Deep Learning* is an approach that is inspired from biological learning [12], which is a representation learning method that represents the data in multiple layers of representation, which all transform the data at one level [80], that was first investigated in 1980s with the discovery of backpropagation by various groups [94–97]. However, Deep Learning methods were initially written off by Machine Learning communities as it was considered to be infeasible to learn anything useful without already having extensive existing knowledge [80]. Deep Learning gained its momentum back in 2006, with several research groups determined that it was possible to pre-train and initialize the models with sensible weights and then fine-tuned the original backpropagation method to obtain models that perform remarkably well, even under sparse data conditions [80, 98–103].

3.2.4.1 Multilayer Perceptrons

There are many different techniques of Deep Learning applications, due to availability of different kinds of sampling, feature extracting techniques. Consider an example case where it aims to build a Deep Learning model that tries to classify vehicles from

images as either *cars*, *planes* or *ships*, which will be the labels. As the first step, a large dataset of car, plane and ship pictures should be compiled [80]. Upon showing the machine one of these pictures, by simply performing linear algebra, it forms a prediction vector, containing one probability score for each label, and the label that has the highest probability, will be the models prediction [80]. Aim is to use an objective function to calculate the error between the desired pattern scores. Depending on the error, models weights are tuned, and this procedure is continued until the training is completed [80]. This type of models are called feed-forward models, meaning the outputs of the model are not fed back to the model [12]. Feed-forward models are one of the most common types of models in Deep Learning applications [80].

One of most basic and quintessential Deep Learning models are the *Multilayer Perceptrons* (MLPs) [12] (Figure 3.3b). MLPs form the basis for many other applications of Deep Learning, such as *Convolutional Neural Network* (CNN) [104], which has wide applications ranging from image or speech recognition to bioinformatics [23]. MLPs require many design choices to deploy, such as *width*, the number of nodes used in a hidden layer, *depth*, the number of hidden layers used, *optimizer*, an iterative algorithm that aims to minimize the cost function; and *activation function*, the functions that are used to calculate the output of a node [12].

3.2.4.2 Stochastic Gradient Descent

Gradient Descent Method [11], forms the basis for all Gradient-Based Optimizers [12]. Gradient descent method obtains a solution for finding the minimum point of the loss function $\Psi(x)$ problem, by making changes to x while trying to minimize the absolute value of the derivative of the loss function $\Psi'(x)$ (Algorithm 2) [12]. Gradient descent may find any type of critical points and not just minimum points. This issue can be addressed however by simply identifying the type of critical point by comparing it to its surroundings and changing the initial estimate if the predicted value is not a minimum.

Almost all Deep Learning optimizers use an extension of the gradient descent method, which is *Stochastic Gradient Descent* (SGD) [12]. Most loss functions that are associated with Machine Learning are additive, determined by taking sum over an entire

Input 1: $f(x)$, the function that we are trying to minimize

Input 2: ϵ , the learning rate hyperparameter

Step 1: Compute $f'(x)$,

Step 2: Adjust x such that $x_{New} = x + \epsilon * \text{sign}(f'(x))$,

Step 3: Repeat steps 1 and 2 until $f'(x)$ becomes 0.

Output: A critical point (local minimum, local maximum, saddle point, global minimum or global maximum) of function $f(x)$

Algorithm 2: Gradient Descent Algorithm [11, 12]

training dataset (Equation 3.29) [12]:

$$\Psi(\theta) = \frac{1}{m} \sum_{i=1}^m l(x_i, y_i, \theta) \quad (3.29)$$

where $\Psi(\theta)$ is the loss function, calculated for model θ , m is the dataset size, and $l(x_i, y_i, \theta)$ is the loss associated with each data pair x_i, y_i for model θ .

Gradient dataset requires application of Equation 3.30 to every single data point, which becomes infeasible as datasets get larger [12]:

$$\nabla_{\theta} \Psi(\theta) = \frac{1}{m} \nabla_{\theta} \sum_{i=1}^m l(x_i, y_i, \theta) \quad (3.30)$$

For this reason, taking uniform mini-batches of data, and estimating expectation based on these mini-batches is much more efficient [12]. Expectation is formed using Equation 3.31, and then used to adjust the model using Equation 3.32 [12]:

$$g = \frac{1}{m'} \nabla_{\theta} \sum_{i=1}^{m'} l(x_i, y_i, \theta) \quad (3.31)$$

$$\theta \leftarrow \theta - \epsilon g \quad (3.32)$$

where g is the expected gradient, m' is the mini-batch size and ϵ is the learning rate.

While SGD is popular, learning using SGD without any modifiers can be slow [12]. Momentum terms, acting as a step-size adjustor, can be introduced to speed up this process (Equations 3.33 and 3.34) [12, 105]:

$$v \longleftarrow \alpha v - \epsilon g \quad (3.33)$$

$$\theta \longleftarrow \theta + v \quad (3.34)$$

where α is a hyperparameter that determines the rate of decay of contributions of previous iterations and takes a value between 0 and 1.

3.2.4.3 ADAM and ADAMax

In almost all cases, some parameters have very high impact on labels while some parameters have almost none, and using the same learning rate ϵ hyperparameter for all parameters consequently leads it having a significant impact on models performance, which results in learning rate a very hard hyperparameter to tune [12]. Usage of momentum introduces a way of vary the step-size, however it also introduces another hyperparameter α to be tuned, while doing so [12]. *Adaptive Moment Estimation* (ADAM), is an SGD-based optimization algorithm that computes adaptive learning rates for various parameters from estimates of first and second moments of the gradients [13]. The algorithm of ADAM is given in Algorithm 3. Recent studies have shown that ADAM is one the state-of-art optimizers and can perform well for variety of classification problems [106].

ADAM operates by scaling gradients of individual weights inversely proportional to a scaled L^2 norm of their prior and current gradients [13]. This L^2 norm can be generalized as L^p (Equations 3.35 and 3.36) [13]:

$$\begin{aligned} v_t &= \beta_2^p v_{t-1} + (1 - \beta_2^p) |g_t|^p \\ &= (1 - \beta_2^p) \sum_{i=1}^t \beta_2^{p(t-i)} |g_i|^p \end{aligned} \quad (3.35)$$

$$\theta_t = \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\widehat{v}_t^{(1/p)} + \epsilon) \quad (3.36)$$

However, for higher values of p such variants tend to be unstable. As an exception to this case, when $p \rightarrow \infty$ (Equations 3.37), a stable algorithm emerges (Algorithm 4), called ADAMax [13]:

Input 1: $f(x)$, the function that we are trying to minimize

Input 2: ϵ , initial learning rate hyperparameter

Input 3: β_1 and β_2 , the exponential decay rate hyperparameters

Step 1: Initialize first moment m , second moment v , and timestep t :

$m_0 \leftarrow 0, v_0 \leftarrow 0, t \leftarrow 0$

Step 2: Apply the following algorithm:

while θ_t is not converged **do**

$t \leftarrow t + 1;$
 $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1});$
 $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t;$
 $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2;$
 $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t);$
 $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t);$
 $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon);$

end

return θ_t

Output: Resulting model θ_t

Algorithm 3: ADAM Algorithm [13]

Input 1: $f(x)$, the function that we are trying to minimize

Input 2: ϵ , initial learning rate hyperparameter

Input 3: β_1 and β_2 , the exponential decay rate hyperparameters

Step 1: Initialize first moment m , infinity norm u , and timestep t :

$m_0 \leftarrow 0, u_0 \leftarrow 0, t \leftarrow 0$

Step 2: Apply the following algorithm:

while θ_t is not converged **do**

$t \leftarrow t + 1;$
 $g_t \leftarrow \nabla_{\theta} f_t(\theta_{t-1});$
 $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t;$
 $u_t \leftarrow \max(\beta_2 \cdot u_{t-1}, |g_t|);$
 $\theta_t \leftarrow \theta_{t-1} - (\alpha / (1 - \beta_1^t)) \cdot m_t / u_t;$

end

return θ_t

Output: Resulting model θ_t

Algorithm 4: ADAMax Algorithm [13]

$$\begin{aligned}
u_t &= \lim_{p \rightarrow \infty} (v_t)^{(1/p)} = \lim_{p \rightarrow \infty} \left((1 - \beta_2^p) \sum_{i=1}^t \beta_2^{p(t-i)} |g_i|^p \right)^{(1/p)} \\
&= \lim_{p \rightarrow \infty} (1 - \beta_2^p)^{(1/p)} \left(\sum_{i=1}^t \beta_2^{p(t-i)} |g_i|^p \right)^{(1/p)} \\
&= \lim_{p \rightarrow \infty} \left(\sum_{i=1}^t (\beta_2^{t-i}) |g_i|^p \right)^{(1/p)} \\
&= \max(\beta_2^{t-1} |g_1|, \beta_2^{t-2} |g_2|, \dots, \beta_2 |g_{t-1}|, |g_t|)
\end{aligned} \tag{3.37}$$

Result of formula in Equation 3.37 is equivalent to the recursive formula shown in Equation 3.38 [13]:

$$u_t = \max(\beta_2 \cdot u_{t-1}, |g_t|) \tag{3.38}$$

3.2.4.4 Binary Cross-Entropy

By reducing its gradient to 0, optimizers aim to minimize an objective function, which is selected based on the nature of the task. While trying to predict TFBSs, task is a binary classification problem, where input sequence is classified as either "*True*" or "*False*". Cross-Entropy, which measures the difference between two different probability distributions, is a widely used Machine Learning loss function (Equation 3.39) [79]:

$$H(x, \theta) = - \sum_{c=1}^C p(x \in c) \times \ln(q(x \in c)) \quad (3.39)$$

where $H(x, \theta)$ is the loss function with parameters of a single data point x and a given model θ . While $p(x_c)$ is expected probability of x in class c , is 1 for its label and 0 for all other labels, and $q(x_c)$ is the probability of x being in class c predicted by the model.

For binary classification, Equation 3.39 can be expanded to get the Equation 3.40 [79]:

$$H(x, \theta) = -p(x \in 0) \times \ln(q(x \in 0)) - p(x \in 1) \times \ln(q(x \in 1)) \quad (3.40)$$

where $p(x \in i)$ is the true probability of class i containing x (either 1 or a 0), and the $q(x \in i)$ is the predicted probability of class i containing x by the model.

For binary classification problems, logistic regression can be used to fit a line to the data that can separate the two classes in the best way possible [107]. Logistic regression tries to fit the data to a sigmoid function (Equation 3.41), which produces a value between 0 and 1, which can be interpreted as the probability of a data point belonging to class 1 [79]:

$$S(x) = \frac{1}{1 + e^{-\theta \cdot x}} = q(x_1) \quad (3.41)$$

Once the probability of the data having a "*True*" label is determined with the sigmoid

function, the probability of a "False" can be determined by Equation 3.42 [79]:

$$q(x_0) = 1 - q(x_1) \quad (3.42)$$

Plugging these values in Equation 3.40, and summing the entropy over an entire training dataset gives us Equation 3.43, which is a loss function that we can apply gradient descent to solve for an optimal model:

$$H(X, \theta) = \sum_{i=1}^X \left(-p(x \in 0) \times \ln\left(1 - \frac{1}{1 + e^{-\theta \cdot x_i}}\right) - p(x \in 1) \times \ln\left(\frac{1}{1 + e^{-\theta \cdot x_i}}\right) \right) \quad (3.43)$$

3.2.4.5 Rectified Linear Unit

In a MLP, all nodes in a hidden layer have some number of inlet links, which feed results of the previous layer to the current layer, each having different weights [79]. Input of a node is the weighted sum of this inlet links (Equation 3.44) [79]:

$$in_j = \sum_{i=0}^n w_{i,j} a_i \quad (3.44)$$

where a_i is the result of the node i , and $w_{i,j}$ is the weight of the link connecting nodes i and j .

Result of this sum is then fed to an activation function g (Equation 3.45), which determines the output of the current node:

$$\begin{aligned} a_j &= g(in_j) \\ &= g\left(\sum_{i=0}^n w_{i,j} a_i\right) \end{aligned} \quad (3.45)$$

In modern neural networks, default activation function suggestion is to use *Rectified Linear Unit* (ReLU) [108–110], definition of which is shown in Equation 3.46 [12]:

$$g(in_j) = \begin{cases} in_j & \text{if } x > 0 \\ 0 & \text{else} \end{cases} \quad (3.46)$$

There are a number of other activation functions based on ReLU, such as Leaky ReLU, a variant of ReLU that allows for a small leakage when input is negative (Equation 3.47) [111], and *Exponential Linear Unit* (ELU), a variant which centers mean around 0 to get better and faster results (Equation 3.48) [112]:

$$g(in_j) = \begin{cases} in_j & \text{if } x > 0 \\ 0.01in_j & \text{else} \end{cases} \quad (3.47)$$

$$g(in_j) = \begin{cases} in_j & \text{if } x > 0 \\ \alpha(e^{-in_j} - 1) & \text{else} \end{cases} \quad (3.48)$$

where α is a hyperparameter to be tuned.

3.2.5 Data Features

When training Machine Learning algorithms, feature extraction must be performed to encode raw data into usable inputs. Due to this, when working with TF binding specificity data, features that can represent sequence specifics of the TFBS needs to be considered.

3.2.5.1 k-mer Dinucleotide Frequency

The effects higher-order k-mers on TFBS binding was a case of interest since the introduction of DWMs (Section 3.2.1). Mordelet et al. [113] has shown that utilizing k-mer dinucleotide frequency feature, more complex and accurate Machine Learning models can be developed. Using this feature, the frequencies of all possible k-length sequence combinations within a given sequence are computed using Equation 3.49, which is then used to form an input array [113]:

$$f_K(X_i, X_j, X_k, \dots) = \frac{N(X_i, X_j, X_k, \dots)}{L} \quad (3.49)$$

where, $X_{i,j,k,\dots} \in \{A, C, G, T\}$, f_K is the frequency of the K-mer, N is the number of occurrences of the subsequence within the 8-mer sequence, and L is the length of the represented sequence.

3.2.5.2 k-Spaced Nucleotide Pair Frequency

Originally proposed and successfully used as a protein sequence encoding tool [114, 115], using k-spaced nucleotide pair frequency to encode DNA sequences has significantly improved model performance when trying to predict N^6 -methyladenine sites [116]. Recent studies also showed that it can be used as a tool to classify TFBSs [34], k-spaced nucleotide pair frequency is a measure of the frequency of nucleotides that are distanced k from each other (Equation 3.50) [116]:

$$f(X_i, X_j, K) = N(X_i, X_j, K)/(L - K - 1) \quad (3.50)$$

where, $X_{i,j} \in \{A, C, G, T\}$, $f(X_i, X_j, K)$ and the $N(X_i, X_j, K)$ is the frequency and the number of observations for the case of the two nucleotides X_i and X_j separated by K nucleotides, while L is the sequence length.

3.2.5.3 Nucleotide Chemical Property

Bari et al. [1] has proposed that the chemical properties of the nucleotides, hydrogen bond strengths (weak or strong), functional groups (amino or keto), and ring structure properties (purine or pyrimidine) can be utilized to encode and classify DNA sequences. Chen et al. [117] has applied this concept to determine N^4 -methylcytosine sites successfully. Wang et al. [34] has applied this concept to their highest affinity TFBS predictor with success, which has showed the promise of the feature when trying to classify TFBSs. A nucleotide chemical property matrix is generated by the use of Table 3.1, where an encoding column is shown for each nucleotide. The se-

Table 3.1: Chemical properties of nucleotides [1]

Nucleotide	Hydrogen Bond	Functional Group	Ring Structure	Encoding
A	Weak (1)	Amino (1)	Purine (1)	(1, 1, 1)
C	Strong(0)	Amino (1)	Pyrimidine (0)	(0, 1, 0)
T	Weak (1)	Keto (0)	Pyrimidine (0)	(1, 0, 0)
G	Strong (0)	Keto (0)	Purine (0)	(0, 0, 0)

quence is encoded by taking the encoding vectors of each nucleotide and putting them together [1].

3.2.5.4 Pseudo-k-tuple Nucleotide Composition

Pseudo-k-tuple nucleotide composition (PseKNC) is a feature that was first developed to compute DNA methylation sites [118]. Structural properties of the DNA has a huge impact on regulatory sites, and usage of PseKNC allows one to account for these properties [118]. Wang et al. [34] has also showed in their study that this feature can be used as an excellent tool for TFBS classification. PseKNC is employed by the usage of Equation 3.51:

$$P_{i,\lambda} = \sum_{n=1}^{L-\lambda-1} (p_i(n) - \bar{p}_i)(p_i(n + \lambda) - \bar{p}_i) \quad (3.51)$$

where i is the index of the physical structural parameter, n is the index of the nucleotide pair in consideration, λ is the pseudo-composition, $p_i(n)$ is the contribution of the n th nucleotide pair to the physical structural parameter in Table 3.2, and $\bar{p}_i$ the mean for all elements in the i th physical structural parameter.

3.2.5.5 Electron-Ion Interaction Pseudopotentials of Trinucleotide

Electron-ion interaction pseudopotentials of trinucleotide (PseEIIP) feature was proposed by Nair and Sreenadhan [3], which has proven to be an efficient DNA encoding tool [119]. Due to being easy to use/understand, and being highly computa-

Table 3.2: Contributions of the nucleotide pairs to physical, structural properties [2]

Step	Twist	Tilt	Roll	Shift	Slide	Rise
AA	0.026	0.038	0.02	1.69	2.26	7.65
AC	0.036	0.038	0.023	1.32	3.03	8.93
AG	0.031	0.037	0.019	1.46	2.03	7.08
AT	0.033	0.036	0.022	1.03	3.83	9.07
CA	0.016	0.025	0.017	1.07	1.78	6.38
CC	0.026	0.042	0.019	1.43	1.65	8.04
CG	0.014	0.026	0.016	1.08	2.00	6.23
GA	0.025	0.038	0.020	1.32	1.93	8.56
GC	0.025	0.036	0.026	1.20	2.61	9.53
TA	0.017	0.018	0.016	0.72	1.20	6.23

tionally efficient, this feature was utilized in many different tasks on trying to predict regulatory regions such as enhancers [120], nucleosomes [121] or even protein hot-spots [122, 123]. PseEIIP is used by computing the 3-mer frequencies in a sequence and taking Hadamard product ($\odot$), by multiplying each frequency with its corresponding electron-ion potential and using resulting elements to form a vector (Equations 3.52 and 3.53) [3]:

$$EIIP_{X_i, X_j, X_k} = EIIP(X_i) + EIIP(X_j) + EIIP(X_k) \quad (3.52)$$

$$V_{X_i, X_j, X_k} = f_{X_i, X_j, X_k} \odot EIIP_{X_i, X_j, X_k} \quad (3.53)$$

where, $X_{i,j,k} \in \{A, C, G, T\}$, EIIP is the electron-ion interaction pseudopotential, f_{X_i, X_j, X_k} is the occurrence frequency of the 3-mer combinations within the sequence, and V_{X_i, X_j, X_k} is the value assigned to the feature vector for 3-mer $X_i X_j X_k$.

The values of EIIP for single nucleotides are given in Table 3.3.

Table 3.3: Electron-ion interaction pseudopotential contributions for the nucleotides [3]

Nucleotide	A	C	G	T
EIIP	0.1260	0.1340	0.0806	0.1335

CHAPTER 4

PHYLOGENETIC FOOTPRINTING METHOD FOR PREDICTING CIS-ACTING DNA ELEMENTS

4.1 Introduction

The aim of this work is, first, to establish the *S. cerevisiae* cADSs curation pipeline (Sc-cADSs-CP). Next is to predict the conserved cADSs in *P. pastoris* by the information flow from Sc-cADSs-CP through pairing cross-species alignments and identify the master TFs that regulate the expressions of the structural genes of central metabolic pathways in *P. pastoris* (Figure 4.1).

In this chapter, a phylogenetic footprinting algorithm (Algorithm 5) aiming to predict *P. pastoris* TFBSs is introduced, and the PWMs for each predicted TFBSs using *S. cerevisiae* as model yeast, are presented.

4.2 Methods

4.2.1 Identification of *S. cerevisiae* and *P. pastoris* Promoters

In the first two steps of the Algorithm 5, *S. cerevisiae* and *P. pastoris* promoters are identified. For the reference host *S. cerevisiae*, UniProt [51] and NCBI [52] databases were used to determine the genes encoding the enzymes and isoenzymes and their subunits involved in the central pathways. The enzymes of *P. pastoris*, which are orthologous to the enzymes of *S. cerevisiae* were determined by protein-protein BLAST protocol provided by NCBI [124]. To calculate the alignment scores, BLOSUM62 scoring matrix was used. The *e*-value (Equation 3.6), is defined as the

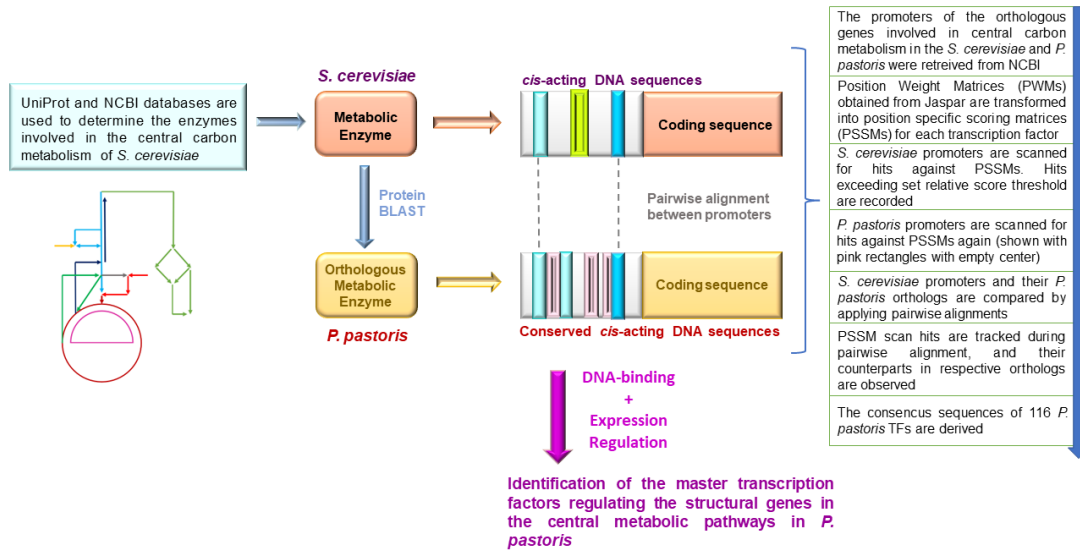


Figure 4.1: A flowchart representation of Footprinting Algorithm

alignment counts that would achieve a score equal or better than the given alignment in a database of same size that has no homologous proteins. For the case of this study, to consider a protein pair homologous, the alignment between them is expected to achieve 10^{-10} *e*-score or lower.

The promoters of the genes of interest of both yeasts were retrieved from the genome sequences as follows: if the distance between the transcription start site of the transcribed gene to the upstream gene is (a) longer than 1500 bp, the promoter length was considered as 1500 bp; (b) shorter than 1000 bp, the promoter length was considered as 1000 bp; (c) between 1000-1500 bp, the promoter length was considered as it is.

4.2.2 Motif Scanning and Pairwise Alignment using Biopython

To annotate the TFBS regions on reference promoter, PWM scan algorithm was used, which is showcased in Algorithm 6 (Section 3.1.1). For *S. cerevisiae*, the TF frequency matrices were gathered from the Transfac® database [6]. If there were more than one frequency matrix for a TF the longest one is used for the calculations. For the PSSM calculations, the background probabilities for C and G are taken as 19% as GC content of *S. cerevisiae* is 38%.

Input: ID of the targeted gene

Step 1: Determine protein sequence and promoter sequence

Step 2: Find the homologous *S. cerevisiae* gene of the target gene by applying protein-protein blast to be used as reference

Step 3: Annotate the TFBS locations on the reference gene's promoter sequence.

Step 4: By comparing the target genes promoter sequence with reference genes promoter sequence, determine conserved regions

Step 5: Annotate the TFBS locations on the target gene's promoter sequence by determining the conserved regions that are annotated in reference gene's promoter sequence.

Output: Annotated TFBS locations over target gene and reference gene.

Algorithm 5: Phylogenetic Footprinting Algorithm

We processed all the frequency matrices of the TFs with the following procedure. For each TF, cut-off values were determined by scanning for hits in exon ≥ 2 sequences of *S. cerevisiae* and allowing a maximum of 3 hits per 10,000 bp when the sequences that achieve a lower relative score than the set cut-off value are discarded (Section 3.1.1). Promoters of *S. cerevisiae* and *P. pastoris* were scanned for TFBS motifs with the Biopython's motifs submodule [125].

In the final step in Phylogenetic Footprinting approach, ScanAlgo, a divide and conquer type algorithm to scan DNA motifs for pairing cross-species alignments, was developed, which is detailed in Algorithm 7. ScanAlgo conceptually was adapted from the Needleman-Wunsch algorithm [10]. ScanAlgo takes homologous promoter sequences for two species as input and produces the putative binding sites predicted using the scan and pairwise alignment functions of the Biopython module [125]. Match score and mismatch penalty were assigned using the EDNAFULL scoring matrix. Affine gap penalty function was utilized, with ten open gap penalties and 1 extend gap penalty.

4.2.3 Performance Analysis

From annotated binding sites of each TF, their *P. pastoris* frequency matrices were developed. These matrices were used to prepare another set of PSSMs, which were then used to scan for TFBSs in *P. pastoris* promoters. The results were compared with the experimental data available in the literature to assess the performance of the model.

4.3 Results

4.3.1 Homologous Proteins of *S. cerevisiae* and *P. pastoris*

After determining the *S. cerevisiae* enzymes that contribute to the central metabolism from NCBI database, using BLAST Protocol, I have identified the homologous *P.*

Input 1: Promoter Sequence

Input 2: PSSM Motif

$Score = 0$ $Score_{MAX} = 0$ $Score_{MIN} = 0$

```
for  $i = 0$  to  $promoterLength - motifLength + 1$  do
     $checkedSequence \leftarrow promoterSequence[i : i + motifLength]$ 
    for  $j = 0$  to  $checkedSequenceLength$  do
         $Score = Score + PSSM(j, checkedSequence[j])$ 
         $Score_{MIN} = Score_{MIN} + \min(PSSM(j))$ 
         $Score_{MAX} = Score_{MAX} + \max(PSSM(j))$ 
    end
    // Apply equation 4 to find the relative score.
     $Score_{REL} = \frac{Score - Score_{MIN}}{Score_{MAX} - Score_{MIN}}$ 
    if  $Score_{REL} \geq Threshold$  then
         $annotatedRegionLocations \leftarrow i$ 
    end
end
```

Output: annotatedRegionLocations

Algorithm 6: PWM Motif Scanning

Input 1: Reference Promoter Sequence

Input 2: Target Promoter Sequence

Input 3: Annotated TFBS locations list for Reference Promoter Sequence

Step 1: Use Needleman-Wunsch pairwise alignment algorithm to align the sequences

Step 2: For each annotated TFBS on reference sequence, find aligned counterpart over the target sequence

Step 3: Check to see if 66% of the TFBS sequence is conserved, if so, annotate on target sequence.

Algorithm 7: ScanAlgo Algorithm

pastoris and *S. cerevisiae* enzymes. Table 4.1 shows an example BLAST for the alcohol dehydrogenase 2 (ADH2) enzyme.

As can be seen in Table 4.1, there are five proteins that can be considered homologous to ADH2, one of which is XP_002491382.1, described as the Mitochondrial alcohol dehydrogenase isozyme III in NCBI database [52], shows an improbable level of similarity and achieved an *e*-Score value of 3e-165. As such, it was considered as the main homologous counterpart of ADH2 protein of *P. pastoris*. XP_002491382.1 is synthesized by PAS_chr2-1_0472 gene in *P. pastoris*. From this, one can deduce that PAS_chr2-1_0472 gene in *P. pastoris* and ADH2 gene in *S. cerevisiae* are homologous.

The results obtained by repeating this procedure for all the proteins that take part in the central metabolism of *S. cerevisiae* are presented in Appendix A.

4.3.2 Annotation of TFBSs in *S. cerevisiae* Promoters

After determining the homologous genes, the promoters of the predicted homologous genes are extracted. The 3rd step of Algorithm 5 is the annotation of TFBS locations on the reference genes' promoter sequences. *S. cerevisiae* genes' promoters were scanned utilizing PWM data of Transfac. A sample result, where P_{ADH2} , promoter

Table 4.1: NCBI BLAST Results of *S. cerevisiae* ADH2 enzyme against *P. pastoris* *GS115* database [4]

Locus ID	Score	Cover	Identity%	E-Score
XP_002491382.1	463	99%	73.78%	3e-165
XP_002492217.1	103	90%	28.83%	8e-26
AOA70192.1	99	90%	28.22%	6e-24
XP_002490014.1	92.8	98%	28.25%	1e-21
XP_002494014.1	65.5	98%	27.52%	2e-12

Table 4.2: Information of Cat8 hits on P_{ADH2} promoter

TF	Location	Strand ¹	Score	Rel. Score	Sequence
Cat8	-479	-	11.065	0.838	TCCGTCTCTCCG
Cat8	-308	-	6.237	0.728	GCCGGAACACCG
Cat8	-306	+	8.253	0.774	CGGAACACCGGG

¹ "+" stands for main sequence, while "-" is for reverse complement.

of ADH2, is scanned for the TF Cat8 motif is shown in Table 4.2 and Figure 4.2.

4.3.3 Annotation of TFBSs on *P. pastoris* Promoters

After scanning *S. cerevisiae* promoters, *S. cerevisiae* and *P. pastoris* promoters were aligned via Needleman-Wunsch pairwise alignment procedure. Next, the annotated regions on *S. cerevisiae* and their aligned counterparts were checked one-by-one to predict and annotate conserved TFBSs on *P. pastoris* promoters. In Figure 4.3, a sample case of TFBS annotation using pairwise alignment is displayed. The two promoter sequences were aligned; and, the Msn2 TFBS and the region aligned were inspected. Since there was one nucleotide deleted in position 3, one nucleotide from both sides were considered to be possible new aligned region candidates. For the second pairwise alignment, the region between 564th and 572th nucleotides of *P. pastoris* has 6 out of 9 conserved nucleotides, which satisfies the conservation criterion of 66%.

Hence, Msn2 TFBS can be considered as conserved on $P_{PAS_chr2-1_0472}$ and was annotated.

1	AATGGCAAACCTGAGCACAACAATACCAGTCCGGATCAACT	40
41	GGCACCATCTCTCCCGTAGTCTCATCTAATTTTTCTTCCG	80
81	GATGAGGTTCCAGATATACCGCAACACCTTTATTATGGTT	120
121	TCCCTGAGGGAATAATAGAATGTCCCATTCGAAATCACCA	160
161	ATTCTAAACCTGGGCGAATTGTATTTTCGGGTTTGTTAACT	200
201	CGTTCCAGTCAGGAATGTTCCACGTGAAGCTATCTTCCAG	240
241	CAAAGTCTCCACTTCTTCATCAAATTGTGGGAGAATACTC	280
281	CCAATGCTCTTATCTATGGGACTTCCGGGAAACACAGTAC	320
321	CGATACTTCCCAATTCGTCTTCAGAGCTCATTGTTTGTTT	360
361	GAAGAGACTAATCAAAGAATCGTTTTCTCAAAAAAATTAA	400
401	TATCTTAACTGATAGTTTGATCAAAGGGGCAAAACGTAGG	440
441	GGCAAACAAACGGAAAAATCGTTTCTCAAATTTTCTGATG	480
481	CCAAGAACTCTAACCAGTCTTATCTAAAAATTGCCTTATG	520
521	A TCCGTCTCTCCG GTTACAGCCTGTGTAAGTATTAATCC	560
561	TGCCTTTCTAATCACCATTCTAATGTTTTAATTAAGGGAT	600
601	TTTGTCTTCATTAACGGCTTTCGCTCATAAAAATGTTATG	640
641	ACGTTTTGCCCGCAGGCGGGAAACCATCCACTTCACGAGA	680
681	CTGATCTCCTCT GCCGGAACACCGGG CATCTCCAATTAT	720
721	AAGTTGGAGAAATAAGAGAATTTTCAGATTGAGAGAATGAA	760
761	AAAAAAAAAAAAAAAAAAGGCAGAGGAGAGCATAGAAAT	799
800	GGGGTTCACTTTTTTGGTAAAGCTATAGCATGCCTATCACAT	840
841	ATAAATAGAGTGCCAGTAGCGACTTTTTTCACACTCGAAA	880
881	TACTCTTACTACTGCTCTCTTGTTGTTTTTATCACTTCTT	920
921	GTTTCTTCTTGGTAAATAGAATATCAAGCTACAAAAAGCA	960
961	TACAATCAACTATCAACTATTAATACTATATCGTAATACACA	1000

Figure 4.2: Graphical showcase of Cat8 hits on P_{ADH2} promoter.

```

403 TCTT-AACTGATAGTTTGATCA AAGGGGCA--AAACGTA 438
    .||| .||...|. ||||..|||| .|||| .|.|.|.
544 CCTTGCACCCCTC-TTTGGACA AAA-TGGCAGTTAGCATT 580

```

(a)

```

424 AAGGGGCAA-- 432
    ||..||||.
564 AAATGGCAGTT 574

```

(b)

Figure 4.3: (a) A segment from pairwise alignment of P_{ADH2} (top) and $P_{PAS_chr2-1_0472}$ (bottom) promoters. Msn2 TFBS annotated on P_{ADH2} , and the region its aligned against is highlighted in yellow. (b) Alignment of highlighted regions. Final Annotated sequence on $P. pastoris$ is highlighted in green.

4.3.4 Computation of $P. pastoris$ PWMs

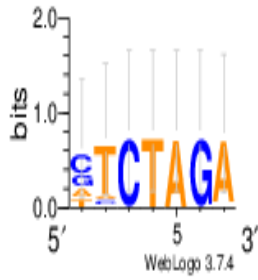
Following the annotation of TFBSs on $P. pastoris$ promoters, PWMs of each TF were computed. In Figure 4.4, a comparison of Abf2 TFBS consensus motifs is presented. In Appendix B, logos for all TFs of $P. pastoris$ are showcased.

4.3.5 Scanning $P. pastoris$ promoters for unidentified TFBSs

While Phylogenetic Footprinting allows for the prediction of conserved TFBSs, it cannot predict the TFBSs on target species that formed after the separation, even if they have exactly the same sequence with another annotated TFBS. For this reason, after computing $P. pastoris$ PWMs, the promoters of the $P. pastoris$ were scanned using these motifs. In Table 4.3, a sample result is provided where annotated TFBSs counts for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7, and Gcr1 TFs for the glycolysis pathway enzymes are showcased. .

4.3.6 Comparison with the Literature

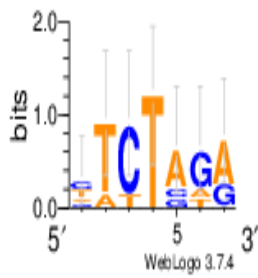
After annotating the TFs, the *S. cerevisiae* TFBSs were compared with the literature using Yeastract [44]. Yeastract database provides the evidences of binding and expression, which were marked on the TFs, by using asterisk (*) or degrees (°) to denote that TF has binding or expression evidence for the given promoter, respectively. The results for all the pathways are presented in Appendix C. For most of the promoters, the results agree with the literature based on the experimental evidence available. However, a significant number of sites that the model predicted in some of the promoters need expression or binding proof experiments, as expected.



(a) *S. cerevisiae* Motif

	1	2	3	4	5	6	7
A	0.18	0.09	0.0	0.0	1.0	0.0	0.95
C	0.39	0.09	1.0	0.0	0.0	0.0	0.02
G	0.25	0.02	0.0	0.0	0.0	1.0	0.02
T	0.18	0.8	0.0	1.0	0.0	0.0	0.02

(b) *S. cerevisiae* PWM



(c) *P. pastoris* Motif

	1	2	3	4	5	6	7
A	4	2	3	1	12	5	14
C	4	3	12	1	3	1	2
G	7	3	1	1	0	11	2
T	5	12	4	17	5	3	2

(d) *P. pastoris* PWM

Figure 4.4: Comparison of *S. cerevisiae* and *P. pastoris* sequence Abf2 TFBS motifs (Generated by WebLogo [5]) and their PWMs (*S. cerevisiae* PWM was accessed from TransFac [6])

Table 4.3: TFBSs counts annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on *S. cerevisiae* and *P. pastoris* promoters that regulate expression of enzymes in the Glycolysis Pathway

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
HXK2 PAS_chr1-4_0561	Cat8(1), Sip4(3), Ert1(4), Mig1(2) ^{°*}	Adr1(3), Cat8(4), Ert1(1), Stb5(3), Msn2(7), Msn4(2), Mig1(2)	—
PGI1 PAS_chapter3_0456	Sip4(1) [°] , Hap2/3/4/5(1), Stb5(2) [°] , Mig1(1), Gcr1(2) ^{°*}	Cat8(2), Ert1(2), Stb5(1), Msn2(3), Msn4(2), Mig1(1)	—
PFK1 PAS_chr2-1_0402	Cat8(1), Sip4(3), Ert1(1), Gcr1(2) [°]	Ert1(4), Stb5(1), Msn2(1), Mig1(1)	—
PFK2 PAS_chr1-4_0047	Cat8(1), Sip4(4), Hap2/3/4/5(1), Ert1(2), Stb5(2), Msn2(2) ^{°*} , Msn4(2)*, Mig1(1), Gcr1(3) [°]	Stb5(1), Msn2(1), Msn4(1), Mig1(1)	Msn4(1)
Continued on next page			

Table 4.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
FBA1 PAS_chr1-1_0072	Adr1(1) ^{°*} , Sip4(2), Rds2(1), Stb5(3), Gcr1(3) [°]	Cat8(2), Ert1(2), Stb5(5), Msn2(8), Msn4(6)	Adr1(1)
TDH2 PAS_chr2-1_0437	Hap2/3/4/5(1), Rds2(1) [°] , Msn2(2) ^{°*} , Msn4(2) [*] , Mig1(2), Gcr1(3) [°]	Cat8(2), Hap2/3/4/5(1), Ert1(3), Msn2(3), Msn4(2), Mig1(4)	Hap2/3/4/5(1)
TDH3 PAS_chr2-1_0437	Hap2/3/4/5(1), Msn2(3) [*] , Msn4(3) [*] , Mig1(3) [°] , Gcr1(2) ^{°*}		
TPI1 PAS_chr3_0951	Cat8(1), Sip4(1), Hap2/3/4/5(1), Msn2(1) [*] , Msn4(1), Gcr1(2) ^{°*}	Adr1(1), Cat8(3), Stb5(3), Msn2(7), Msn4(3), Mig1(3)	Cat8 (1)
PGK1 PAS_chr1-4_0292	Cat8(1) [°] , Sip4(2), Msn2(1) ^{°*} , Msn4(1) [*] , Gcr1(2) ^{°*}	Adr1(1), Cat8(2), Ert1(4), Msn2(1)	—
Continued on next page			

Table 4.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
GPM1 PAS_chr3_0826	Adr1(1), Gcr1(2) ^{°*}	Adr1(1), Cat8(2), Ert1(4), Stb5(1), Msn2(3), Msn4(3)	Adr1(1)
ENO2 PAS_chr3_0082	Sip4(1), Stb5(1) [°] , Mig1(2)*, Gcr1(2) ^{°*}	Cat8(3), Sip4(2), Ert1(1), Stb5(2), Msn2(6), Msn4(4), Mig1(2)	Msn2(1), Msn4(1), Mig1(1)
CDC19 PAS_chr2-1_0769	Adr1(1) [°] , Cat8(2), Sip4(6), Hap2/3/4/5(1), Rds2(2), Ert1(5), Stb5(2), Msn2(3) ^{°*} , Msn4(3) ^{°*} , Mig1(3) [°] , Gcr1(3) ^{°*}	Cat8(2), Sip4(2), Stb5(1), Msn2(5), Msn4(2), Mig1(1)	Sip4(2), Ert1(1), Stb5(1)

[°] Expression evidence was found in literature.

* Binding evidence was found in literature.

CHAPTER 5

NOVEL STRATEGY DESIGNED WITH MACHINE LEARNING ALGORITHMIC MODELS FOR PREDICTING CIS-ACTING DNA SITES

5.1 Introduction

In this chapter, a Greedy Machine Learning Algorithm (Algorithm 8) is presented which aims to model affinities of *S. cerevisiae* TFs towards 8-mers in the best possible way. Performances of the models, and their comparison with the literature are presented.

5.2 Methods

5.2.1 Machine Learning Method

5.2.1.1 Dataset

A database of 274 high-resolution PBM data for 150 for *S. cerevisiae* transcription factors was assembled [45, 48]. E-scores were extracted from PBM data and 8-mer DNA sequences that are recognized by the TFs were determined by setting a threshold value of 99% quantile is recognized, and the rest are discarded.

5.2.1.2 Embedding

8-mer DNA sequences were embedded into arrays by utilizing k-mer dinucleotide frequency, k-spaced nucleotide pair frequency, nucleotide chemical property, PseKNC,

and PseEIIP as feature representation methods, like Wang et al. [34]. Since different TFs may recognize different features, a pool of the best features for modelling were determined for all TFs. This was achieved by training the models first using an array that utilizes all the five features. Then, training procedure was repeated five more times, removing one feature each time. The feature that improved the results the greatest, if dropped (if there are any) is discarded. This procedure is repeated until either removing none of the remaining features improves the results, or there is only one feature left.

Each of the following feature representation methods generates a vector, which were then all concatenated to get the final vector.

1. k-mer dinucleotide frequency

The k-mer dinucleotide frequency feature, representing the frequencies of all k-length sequence combinations within a given sequence, was given by measuring the frequencies of all 1 to 4 length subsequences of a given 8-mer. Measured frequencies were then converted into an array with dimensions $340 (4^1 + 4^2 + 4^3 + 4^4) * 1$.

2. k-spaced nucleotide pair frequency

k was changed from 0 to 4, the frequencies of all possible nucleotide pairs that has a distance 0 to 4 were measured, resulting in an array of shape $80 (4 * 4 * 5) * 1$.

3. Nucleotide chemical property

Feature representation of each nucleotide were added to the input array by transforming them all into 3x1 vectors. The vectors were concatenated into a $24 (3 * 8) * 1$ array.

4. Pseudo nucleotide composition

Six physical, structural properties of the DNA (Twist, Tilt, Shift, Slide, Rise, and Roll) were computed by considering adjacent nucleotide pairs within a given sequence to represent the data, which in turn results in a $42 (7 * 6) * 1$ array.

5. Electron-ion interaction pseudopotentials of trinucleotide

A $64 (4^3) * 1$ array of zeros was prepared, that has a column for every possible 3-mer combination. All 3-mer subsequences within a given 8-mer were considered. Their Electron-ion interaction pseudopotentials were calculated, which were then plugged into the input array.

Upon using all the five features, a data array with $(274*65536*551*1)$ was generated, where 274 is the TF count, 65536 is all possible 8-mer combinations, 551 is width of the array and 1 is the depth of the array. The width of the array was then varied as feature(s) discarded (Section 3.2.5).

5.2.1.3 Model Training

While training the models, each TF was approached as a separate problem, and a greedy approach was used while modelling each individual TF to get the best overall performance.

First, a data array utilizing all the five features is generated. This array was then divided into three, such that 76.5, 8.5, and 15% of the data were used to train, validate, and test the models, respectively. The train dataset was then used to train models using three Machine Learning systems, which are Random Forest, XGBoost, and Neural Networks. Random Forests were implemented using Sci-Kit Learn library for Python, and default hyperparameters were used [126]. XGBoost was implemented using XGBoost Library for Python, and default parameters suggested was used [90]. Finally, Neural Networks were generated using TensorFlow library for Python [127]. For MLPs, for layer count, while it is theoretically possible to achieve any desired non-zero error with just one hidden layer, empirically increasing the number of hidden layers results in improved accuracy, provided that there are enough hidden units for the various tasks [12]. For this reason, one simple Neural Network with one hidden layer, and four MLPs, which range two to five in depth, were generated to find the most optimal depth. While training the Neural Networks, three different optimizers were tested, base SGD trainer, Adam, and Adamax. Also two different activation functions, ReLU and ELU, were checked. During the training of Neural Networks,

for All TFs do

Step 1: Obtain binding True/False classification training and testing data.

Step 2: Train MLPs, XGBoost, Random Forest models on training data.

Step 3: Validate the models on testing data.

Step 4: Pick the one that performs the best on testing data.

end

Step 5: Create a library of models using the best models picked.

Algorithm 8: Greedy Algorithm used to model TF interactions

maximum epoch counter was set to 500, and an early stopping condition was installed, where if the loss does not decrease for 20 Epochs (50 for SGD) training procedure is stopped and the model is rolled back to the best iteration. A hardware with 200 GB SSD memory, 32 GB RAM and 6 core chip was used for training.

After training one set of models using all five features, five more models were trained using the four features, removing each feature once. If one (or more) of the 4-feature models performed better than the 5-features model, the best performing 4-features were picked and four more models were trained, using three feature combinations, removing each feature once. This procedure is repeated until the best feature combination for each TF is achieved. In figure 5.1, a flowchart representation of the algorithm 8, combined with the scanning procedure is given.

5.2.1.4 Statistical Analysis of Models

All the seven models were trained and tested with the same datasets, for all the TFs. Due to the biased nature of the problem, if a model answers 0 ("This TF does not bind to this sequence") to all 8-mers, it would achieve 99.0% accuracy. Due to this, to evaluate model performance, recall, precision, and Mathews' Correlation Coefficient (MCC) metrics were used instead of accuracy:

$$Recall = \frac{TP}{TP + FN} \quad (5.1)$$

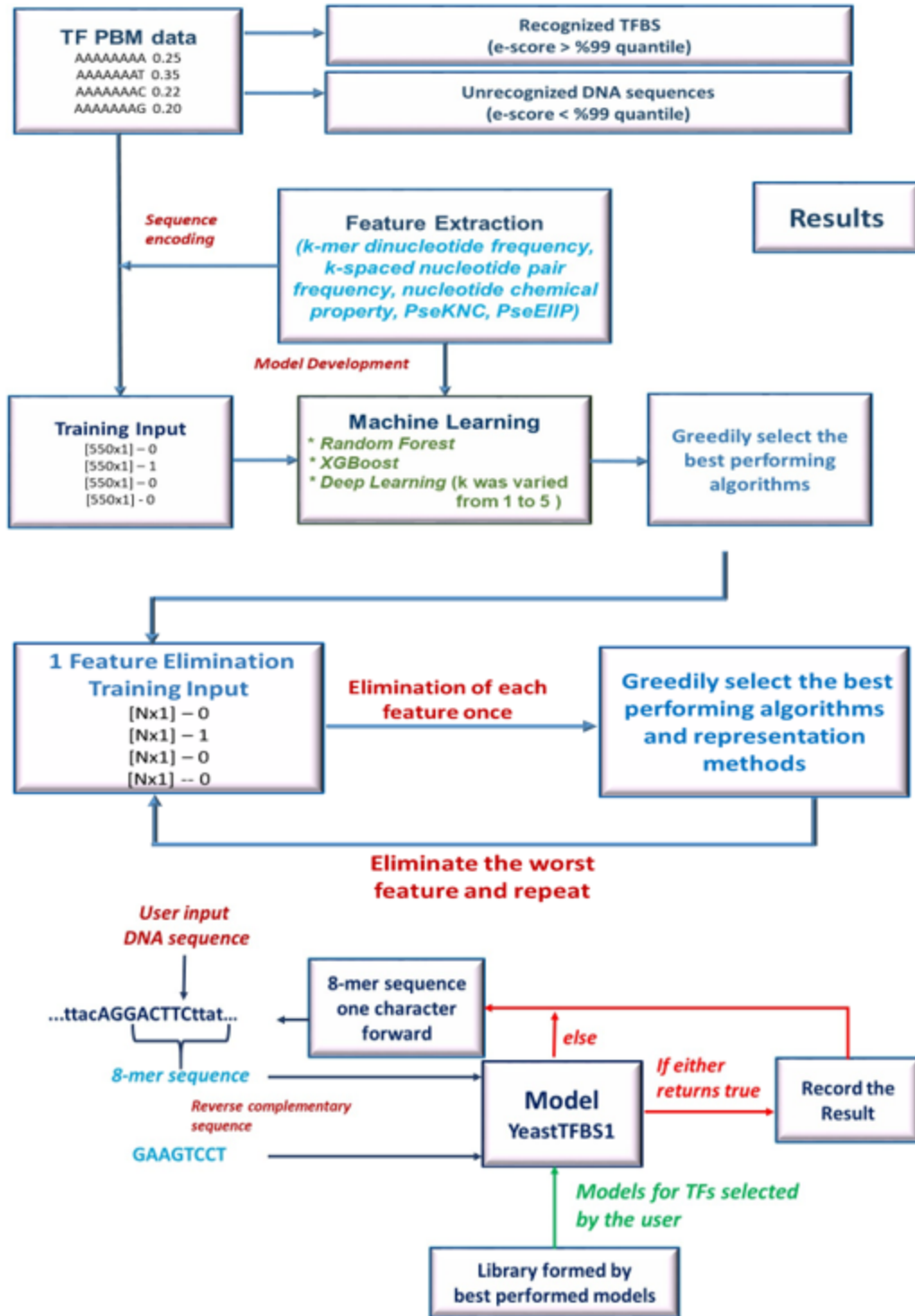


Figure 5.1: A Flowchart of Training and Scanning Algorithms

$$Precision = \frac{TP}{TP + FP} \quad (5.2)$$

$$MCC = \frac{(TP * TN) - (FP * FN)}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}} \quad (5.3)$$

MCC ranges from -1, achieved when true positive and true negative is 0 which indicates a model that predicts all inputs wrongly, to 1, achieved when false positive and false negative is 0 which indicates a model that predicts all inputs correctly. Thus, an increasing MCC indicate a better predicting performance for the models.

5.3 Results

5.3.1 Neural Network Optimization

The optimal hyperparameteres were determined by testing them on a randomly selected 28 TF PBM data from the dataset, which amounts to a subset that is 10% of the original dataset. The results were compared with respect to performance metrics, in addition to other factors such as amount of epochs required and training time.

5.3.1.1 Optimizer Selection

Three optimizers were tested on 28 TFs by training the five models of varying depths, observing the change of average of best MCC scores for each TF, and the average number of epochs required for the training. During the optimizer selection tests, ReLU activation function was used. The results showed that all optimizers achieved close to same MCCs, around 0.66, with ADAMAX being slightly higher compared to the rest (Table 5.1). However, upon checking required epoch counts to converge, it can be clearly observed that SGD requires too many epochs to be feasible with large scale processes, and the optimizers with adaptive learning rates are more advantageous in this regard. While not as bad as SGD, ADAMAX's convergence rate is slower compared to ADAM's, which converges within 13 epochs on average, almost 1/3 of the number of epochs required with ADAMAX. On the basis of these results, I conclude that, considering 1 epoch takes 1 second approximately on average, for

Table 5.1: Optimizer Performances upon training models for 28 TFs using ReLU activation Function

Optimizer	Avg. Best MCCs	Avg. # of Epochs
SGD	0.662	424.86
ADAM	0.662	12.99
ADAMax	0.665	35.74

8220 models (30 for all 274 TFs), using ADAM can save up to 45 hours. Therefore, ADAM was selected as the main optimizer.

5.3.1.2 Activation Function Selection

For activation function selection, ADAM was used with the activation functions ReLU, ELU and Leaky ReLU. The test results (Table 5.2) shows clearly that ReLU is the inferior option, with being the slowest and most inaccurate. ReLU's performance can be due to the lack of negative values, unlike ELU and Leaky ReLU. The evidence indicates that, TFs binding affinity may be affected negatively from some certain features of the DNA sequence. These interactions are disregarded with ReLU, which may be the reasoning behind poor performances experimented using ReLU, compared to other activation functions. The superior activation functions are ELU and Leaky ReLU, and their performances are similar to each other with an 0.0028 average MCC difference between their best performing models. The convergence time difference between the two is 1.5 epochs, that corresponds to 3 hours in large scale. Therefore, I conclude that, 0.0028 MCC difference is not as significant as the time difference that would occur in larger scale tests; and consequently, ELU was selected as the main activation function.

5.3.2 Training Results

After determination of superior Deep Learning optimizer (ADAM) and the activation functions (ELU and Leaky ReLU), for each TF PBM data, five feedforward

Table 5.2: Activation Function Performances upon training models for 28 TFs using ReLU activation Function

Function	Avg. Best MCCs	Avg. # of Epochs
ReLU	0.662	12.99
ELU	0.667	7.87
Leaky ReLU	0.670	9.36

Table 5.3: Performance Metrics achieved by three Machine Learning Methods over 274 TFs

Method	Precision	Recall	MCC
XGB	0.896	0.666	0.765
DL	0.626	0.770	0.682
RF	0.871	0.424	0.588
Greedy	0.884	0.679	0.766

neural networks (one single layer perceptron and four MLPs with depths varying between two to five) alongside one XGBoost model and one Random Forest model were trained. The results showcased a clear domination of XGBoost, which remarkably turned out to be the best model for over 254 TFs, averaging 0.765 MCC over all the TFs. Deep Learning is the superior model for only the 24 TFs, but is significantly behind XGBoost on average, achieving 0.682. Random Forest predictions were not satisfactory and could not outperform neither Deep Learning nor XGBoost for any of the TFs with low MCC scores of 0.588. The detailed performance metrics of each Machine Learning method is given in Table 5.3.

The results also showed that the tree-based Machine Learning system, XGBoost, successfully predicts with higher than 0.85 precision. Tree-based Machine Learning Algorithm XGBoost enable high MCC scores despite achieving less TPs compared to the Deep Learning method. The designed greedy approach for selecting the best performing model improves predictions slightly, with an average MCC score of 0.766 over all 274 TFs.

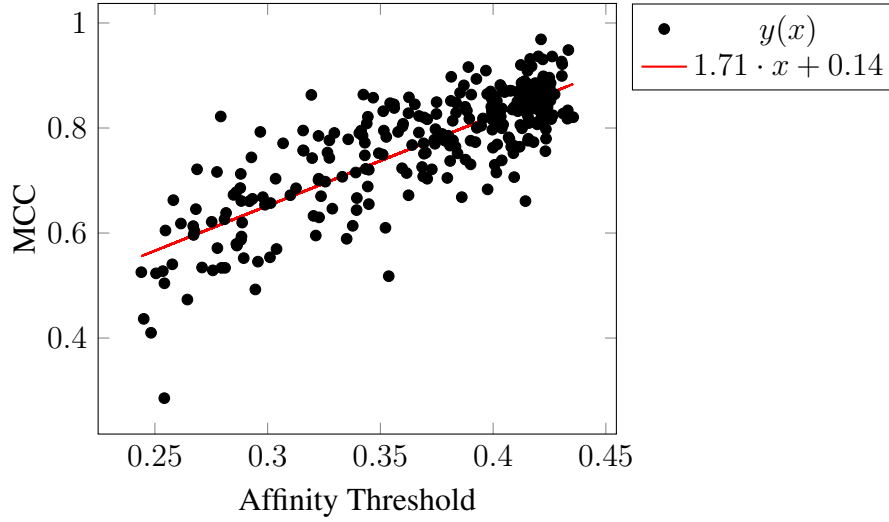


Figure 5.2: Dependency of the Model MCCs to Thresholds

Table 5.4: Performance Metrics of the three subgroups of TFs.

TF Group	Precision	Recall	MCC	# of TFs
HSTF	0.869	0.427	0.599	47
MSTF	0.875	0.682	0.766	124
LSTF	0.903	0.791	0.842	103

High specificity TFs require more complex conditions to bind, which only a sparse group of 8-mers satisfies. Thus, they require more complex models to predict precisely, compared to lower specificity TFs. The phenomena is observed in the results, while good performing models showed a clear bias towards TFs with lower binding specificities (Figure 5.2).

Dividing PBM datasets to three arbitrary categories as *high*, *medium* and *low specificity TFs* (HSTF, MSTF, LSTF), by grouping TFs with thresholds 0.4 or higher as LSTFs, with thresholds 0.3 or lower as HSTFs, with thresholds in between them as MSTFs, performance drop of the models can be further illustrated as shown in Table 5.4.

5.3.3 Feature Selection

In order to increase the prediction performance of the models, the significance and impact of the five features were investigated for each Algorithmic model with a systematic program by eliminating each feature: first i) single-handedly, ii) two features at a time, iii) three features at a time, and iv) four features at a time. If elimination of any one the remaining features did not improve on the results from the previous iteration, the iteration is stopped and, the results and the feature set from the previous iteration is recorded.

5.3.3.1 1st-Order Feature Elimination

After achieving an average 0.766 MCC score with the five features (as indicated with "Reference" line in Figure 5.4), feature elimination procedure was introduced by training five more models per TF, to observe the importance of the feature representation methods. In Table 5.5, change of performance metrics for each Machine Learning method with respect to each eliminated feature, which are coded as "—", 0, 1, 2, 3, 4 for "None", "k-mer Dinucleotide Frequency", "k-spaced Nucleotide Pair Frequency", "Nucleotide Chemical Property", "PseKNC" and "PseEIIP", respectively. Consequently, best case models and best case feature combinations were selected greedily to observe how the changes affect the overall prediction performance.

Some of the models were performed better by elimination of some features. The TFs whose best model has experienced an improvement after a elimination of a specific feature is showcased in Figure 5.3, where a reference line is also provided for number of TFs whose best model did not experience improvements upon elimination of any of the features. A graphical comparison of average MCC scores of superior models with respect to eliminating some of feature(s) is also showcased, in Figure 5.4.

XGBoost predicted with higher MCC scores when either of k-mer Dinucleotide Frequency or Nucleotide Chemical Property is omitted, while eliminating PseKNC significantly reduced the XGBoost performance. Deep Learning models were stable to changing feature dimensions except for eliminating PseKNC, which resulted in a significant loss of performance. In contrast to XGBoost and Deep Learning, Random

Table 5.5: Change of Performance Metrics with respect to removed features

Method	Removed Ft.	Precision	Recall	MCC
XGB	—	0.896	0.666	0.765
	0	0.914	0.676	0.779
	1	0.889	0.650	0.752
	2	0.900	0.674	0.771
	3	0.777	0.594	0.671
	4	0.896	0.666	0.765
DL	—	0.626	0.770	0.682
	0	0.627	0.764	0.682
	1	0.613	0.757	0.666
	2	0.626	0.771	0.682
	3	0.593	0.762	0.657
	4	0.631	0.767	0.683
RF	—	0.871	0.424	0.588
	0	0.905	0.385	0.570
	1	0.873	0.389	0.561
	2	0.865	0.424	0.587
	3	0.842	0.465	0.608
	4	0.878	0.411	0.581
GREEDY	—	0.884	0.679	0.766
	0	0.904	0.688	0.781
	1	0.878	0.666	0.755
	2	0.892	0.684	0.773
	3	0.735	0.667	0.683
	4	0.890	0.674	0.766
GREEDY	GREEDY	0.910	0.707	0.795

Forests experienced the loss of performance by eliminating any of the features except for PseKNC, which improved Random Forest predictions significantly.

For finite sized datasets, Hughes Phenomenon [128] and Curse of Dimensionality

[129] dictate that increasing the number of dimensions after a certain point decline or stabilize the training performances of Machine Learning Algorithms. This phenomenon can be used to explain the behaviour of XGBoost and Deep Learning models when removing k-mer Dinucleotide Frequency, Nucleotide Chemical Property, both of which improved XGBoost prediction performance upon elimination. Consequently, I conclude that these features do not provide enough information on the dataset, that would justify the training performance decline caused by increasing dimension size. k-spaced Nucleotide Pair Frequency improved some of the models performances upon elimination, however, it was rarely the best option, and for most models of XGBoost and Deep Learning, resulted in a decline. PseKNC did improve almost no models, and instead was devastating for performance metrics. One can conclude from these results that PseKNC and k-spaced Nucleotide Pair Frequency features are much more important compared to others, especially PseKNC, which feeds information about the DNAs shape for the 8-mers.

Selecting the best 4-feature set combination improved the prediction performances for HSTFs drastically, reducing the models dependency on high threshold values, as showcased in Figure 5.5 and Table 5.6.

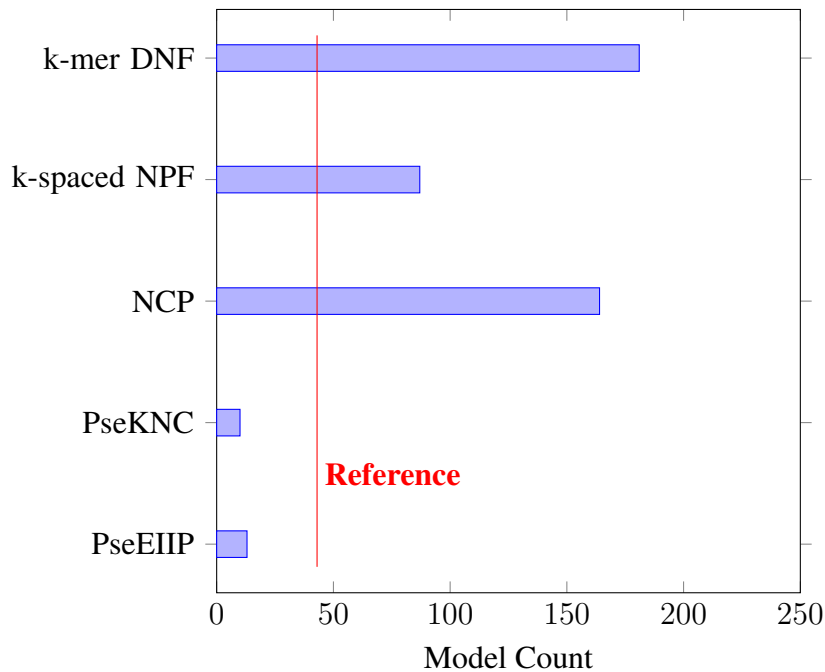


Figure 5.3: Number of Best Models Benefited from Features elimination

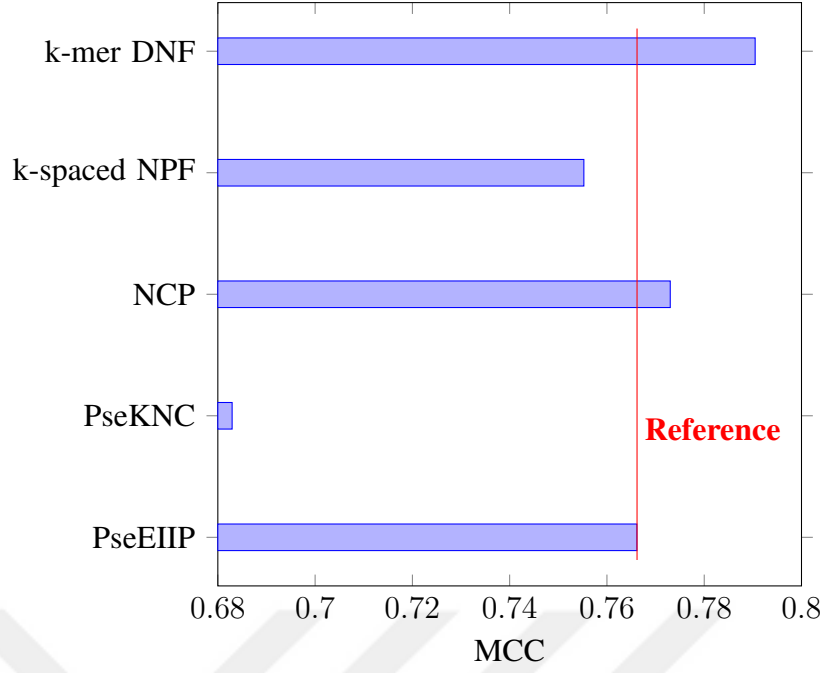


Figure 5.4: Impact of Eliminating Features on Superior Models

Table 5.6: Performance Metrics of the three subgroups of TFs when Feature Pool size is reduced to four

TF Group	Precision	Recall	MCC	# of TFs
HSTF	0.912	0.482	0.654	47
MSTF	0.900	0.704	0.791	124
LSTF	0.921	0.813	0.863	103

Overall, the best performing models for 103 LSTFs performed with a remarkable MCC score of 0.863, while 124 MSTFs performed with an MCC score of 0.791. HSTFs were modeled with an MCC score of 0.654.

5.3.3.2 *N*th-Order Feature Elimination

After eliminating features once from all the TFs, for 230 TFs, prediction performances for the superior models improved, while for 44 TFs did not experienced an improvement in performance. TFs that did not achieve improved results were retired, and the remaining 230 TFs were trained four more times, with 3-feature combina-

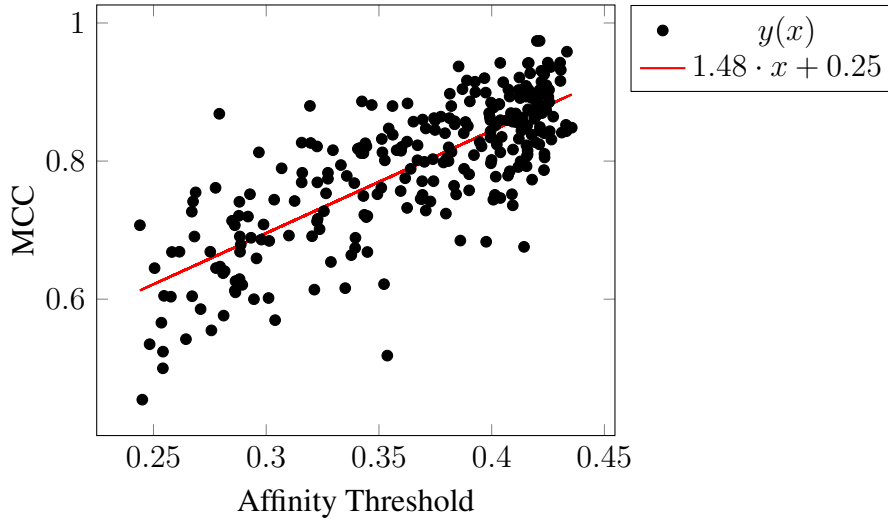


Figure 5.5: Dependency of the Model MCCs to Thresholds when Feature Pool size is reduced to four

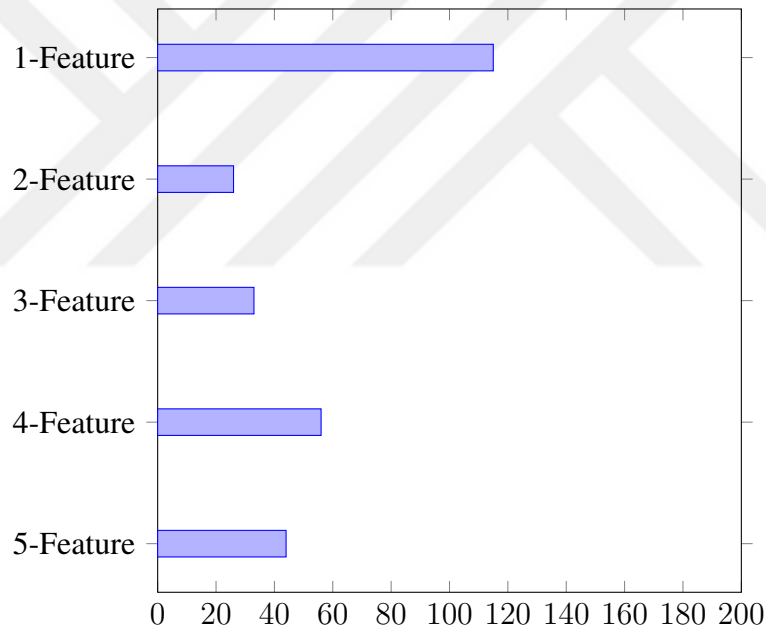


Figure 5.6: Number of superior models for specific TFs with respect to Feature set size

tions. This procedure was repeated until either there are no models that improve the current best model, or feature set size was reduced to 1. A graphical comparison of TFs that were retired in each iteration is given in Figure 5.6.

The performance of HSTFs especially increased as number of features decreased,

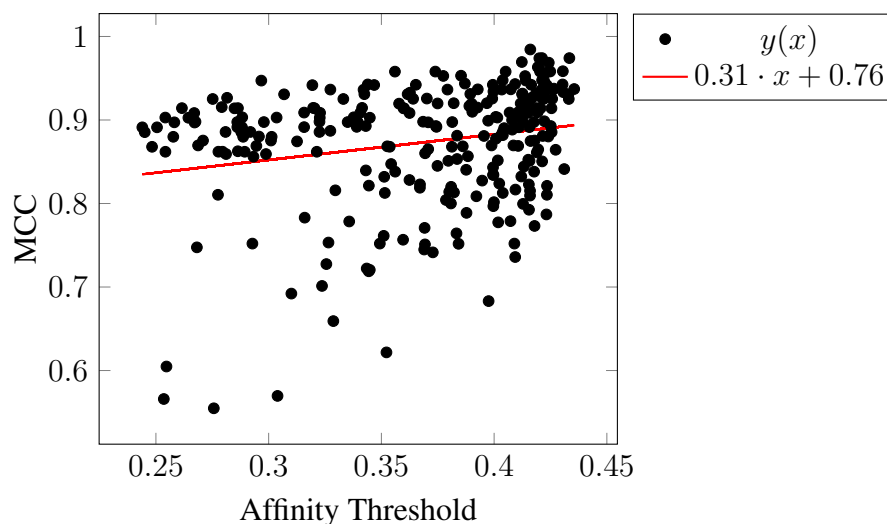


Figure 5.7: Dependency of the Model MCCs to Thresholds when Feature Pool is varied to include best performing features

Table 5.7: Performance Metrics of the three subgroups of TFs when Feature Pool is varied to include best performing features

TF Group	Precision	Recall	MCC	# of TFs
HSTF	0.959	0.782	0.858	47
MSTF	0.933	0.794	0.861	124
LSTF	0.943	0.854	0.896	103

reaching a final average MCC Score of 0.861. Average MCC Score for LSTFs rose to 0.896, and overall average rose to 0.873. The thresholds effect on MCC with best feature sets selected for all TFs also diminished, with slope of Affinity Threshold versus MCC trendline reducing to 0.31, as shown in 5.7. In Appendix D, the effect of eliminating features on the model performances were given in detail for each TF.

The results of this MSc Thesis demonstrated that it is possible to classify high-affinity 8-mers for each TF as an individual problem and the greedily selecting feature pools and modelling tools allows MCC scores of 0.896 for LSTFs, while achieving MCC scores of 0.873 on average, without requiring to filter out the low-affinity binding sites. In the literature, for yeast transcription factor binding site prediction, Wang et al. [34] showed that using XGBoost as a feature selection tool and a classifier

algorithm, manually classified the selected high affinity 8-mers to TF families, using the same five features.

5.3.3.3 Model Stability

The stability of the models was tested by taking training models on randomly selected 28 TFs 10 times each under exactly same conditions. XGBoost and Random Forest were the stable models, returning same results after every iteration. While not as stable, Deep Learning models returns the results with an MCC error margin within ± 1.5 , which was considered acceptable.

5.3.4 Word2Vec Supported Models

To develop a single model that can predict any TF-TFBS relation based on protein sequences and the 8-mer inputs, amino acid sequences of all proteins of *S. cerevisiae* were also extracted from UniProt [51]. The protein sequences were then divided into 4-mer amino acid sequences, which were then embedded into vectors with using Word2Vec Algorithm [35], using the TensorFlow library. Visualization of resulting 153623 unique 4-mer vectors is presented in Figure 5.8.

Each 4-mer vector has 128-dimensions, and the ones closest to 4-mers that they have commonly appear right next to in *S. cerevisiae* proteins (e.g. QYWQ's closes neighbour is QYWW). After determining 4-mer vectors, protein vectors were determined

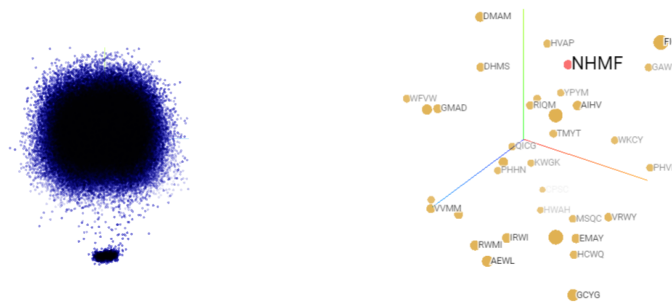


Figure 5.8: PCA Projection of 4-mer aminoacid vectors plotted with TensorFlow Projector tool

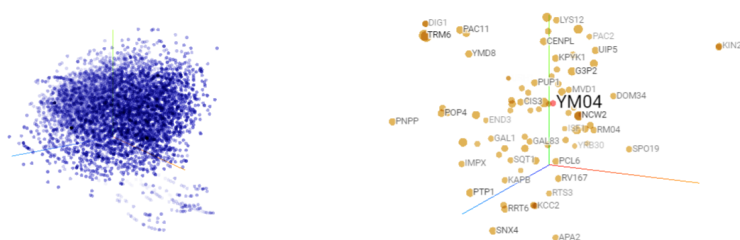


Figure 5.9: PCA Projection of protein vectors plotted with TensorFlow Projector tool

Table 5.8: MCC Scores of Word2Vec included models with changing training hyper-parameters

Act. Func.	Model Depth				
	1	2	3	4	5
ReLU	0.491	0.612	0.558	0.563	0.460
ELU	0.615	0.638	0.678	0.671	0.658

by summing the vectors of all 4-mers that they contain, which were then normalized. The resulting all of *S. cerevisiae* 6062 proteins vectors is shown in Figure 5.9.

In protein vectors, proteins that share large quantities of closely related amino acid sequences appear closer. Since the protein vectors are the sum of 4-mer vectors, they also have 128-dimensions each.

After developing protein vectors, the resulting vectors were combined into 551 dimension TFBS vectors, which resulted in 679-dimension TF-TFBS interaction vectors. Since TF-TFBS interaction vectors no longer require individual models, all PBM datasets available for 150 unique TFs were merged into a single set with 9830400 data points.

10 Deep Learning models were trained with Adam optimizers for small-scale 110 TFs to determine the optimal depth (1 to 5) and activation function (ReLU or ELU). Results showed that ELU with 3-depth is superior to other models (Table 5.8). Thus, was 3-depth was selected to be used in large scale models.

The 3-depth ELU model was then scaled up to larger scale, with 150 TFs. The model

Precision	Recall	MCC
0.727	0.602	0.659

Table 5.9: ELU 3-depth Deep Learning Models performance metrics on 150 TFs

was trained such that the stopping condition was either not achieving any decrease in validation loss for 100 epochs, or reaching the achieving maximum number of epochs, which was set to be 1000. Change of the loss function over epochs is given in Figure 5.10, while achieved models performance metrics is given in Table 5.9.

Word2Vec integrated models performed close to (but slightly worse than) regular Deep Learning models with no feature elimination, which is to be expected, as the model not only tries to predict whether a TF has high affinity to a 8-mer or not, but also tries to predict the TF. While the disadvantage is slightly lower accuracy, the advantage of this approach is the use of one model instead of many, which results in significantly higher disk usage efficiency, reducing the 1.5 GBs occupied of 250 models to just 5 MB. Yet, further improvements are necessary, such as trial of Machine Learning systems other than Deep Learning, to approach the best performances achieved, which is near 0.85 MCC.

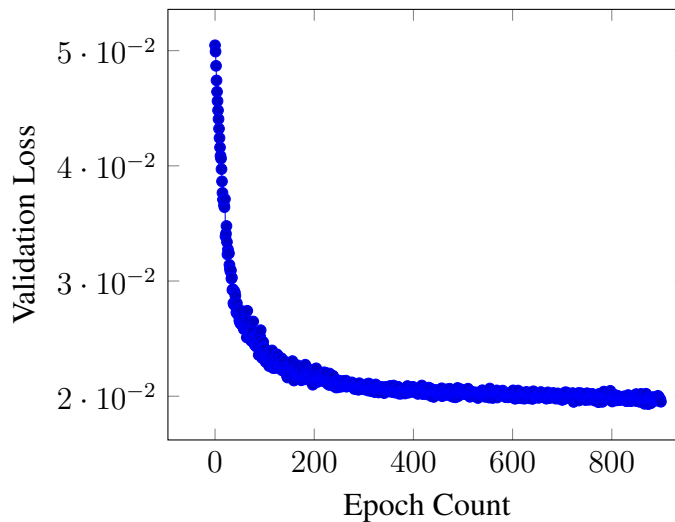


Figure 5.10: Change of Validation Loss during training 3-depth ELU model over 1000 Epochs

CHAPTER 6

CONCLUSIONS

In this thesis, the aim was to develop methodologies for predicting transcription factor binding sites in yeast cells with high accuracy. To this end, two methods were developed. The first, based on the traditional phylogenetic footprinting approach; and, the second is the novel strategy designed with machine learning algorithmic models.

First, the phylogenetic footprinting algorithm was developed and tested, which aims to transform experimentally determined *S. cerevisiae* PWM motifs to predict PWM motifs for *P. pastoris*. This was done by annotating TFBSs on *S. cerevisiae* promoter sequences, and then using pairwise alignment algorithm to observe conserved motifs. Conserved motifs were then recorded, to form new PWMs. Using this methodology, given 58 *S. cerevisiae* and 52 *P. pastoris* promoters, in addition to a database of *S. cerevisiae* TFs that has 182 TFs as inputs, 116 TFs were annotated on *P. pastoris* promoters, PWMs of which were reported.

Next, 5 Neural Networks, 1 XGBoost, and 1 Random Forest model were trained, aiming to predict the high-affinity 8-mers of *S. cerevisiae* TFs. All possible 8 character long DNA sequences were embedded into a 550x1 numerical array, using 5 feature representation methods. After training the models, best training method for each case was selected greedily, as all TFs prioritize different features, and hence, best model might be subject to change depending on the TFs. The results conclusively demonstrated that XGBoost is the superior algorithmic model with MCC scores of 0.765, while with the designed Greedy method the MCC score was 0.766. The results also showed that models predict LSTFs with significantly higher accuracy, compared to HSTFs, where 103 LSTFs were predicted with a MCC scores of 0.842, while 47 HSTFs were predicted with just a MCC scores of 0.599. To observe the importance

of each feature, a second training session was run by omitting removing each feature once. Removal of either k-mer Dinucleotide Frequency or Nucleotide Chemical Property features boosted the model performances significantly, especially for XGBoost models. Upon greedy selection of best features for each TF, overall performance was increased to 0.873, while performance for HSTFs was increased to 0.861. Final models were able to predict LSTFs with 0.896 MCC.

Finally, a single Word2Vec-integrated model that tries to predict any TF-TFBS interaction was also proposed, as exploration of such methodology would allow for significantly reduced disk-space usage, improving the efficiency of the program.



REFERENCES

- [1] A. T. M. G. Bari, M. R. Reaz, H.-J. Choi, and B.-S. Jeong, “DNA Encoding for Splice Site Prediction in Large DNA Sequence,” in *Database Systems for Advanced Applications* (B. Hong, X. Meng, L. Chen, W. Winiwarter, and W. Song, eds.), (Berlin, Heidelberg), pp. 46–58, Springer Berlin Heidelberg, 2013.
- [2] J. R. Goñi, A. Pérez, D. Torrents, and M. Orozco, “Determining promoter location based on DNA structure first-principles calculations,” *Genome Biology*, vol. 8, pp. 1–10, dec 2007.
- [3] A. S. Nair and S. P. Sreenadhan, “A coding measure scheme employing electron-ion interaction pseudopotential (EIIP),” *Bioinformation*, vol. 1, p. 197, oct 2006.
- [4] D. S. Johnson, A. Mortazavi, R. M. Myers, and B. Wold, “Genome-wide mapping of in vivo protein-DNA interactions,” *Science*, vol. 316, pp. 1497–1502, jun 2007.
- [5] G. E. Crooks, G. Hon, J.-M. Chandonia, and S. E. Brenner, “WebLogo: a sequence logo generator,” *Genome research*, vol. 14, pp. 1188–1190, jun 2004.
- [6] E. Wingender, X. Chen, E. Fricke, R. Geffers, R. Hehl, I. Liebich, M. Krull, V. Matys, H. Michael, R. Ohnhäuser, M. Prüß, F. Schacherer, S. Thiele, and S. Urbach, “The TRANSFAC system on gene expression regulation,” *Nucleic Acids Research*, vol. 29, pp. 281–283, jan 2001.
- [7] T. I. Lee, N. J. Rinaldi, F. Robert, D. T. Odom, Z. Bar-Joseph, G. K. Gerber, N. M. Hannett, C. T. Harbison, C. M. Thompson, I. Simon, J. Zeitlinger, E. G. Jennings, H. L. Murray, D. B. Gordon, B. Ren, J. J. Wyrick, J.-B. Tagne, T. L. Volkert, E. Fraenkel, D. K. Gifford, and R. A. Young, “Transcriptional Regulatory Networks in *Saccharomyces cerevisiae*,” *Science*, vol. 298, pp. 799–804, oct 2002.

- [8] P. Jouhten, E. Vartiainen, A. Huuskonen, A. Tamminen, M. Toivari, M. Wiebe, L. Ruohonen, M. Penttilä, and H. Maaheimo, “Oxygen dependence of metabolic fluxes and energy generation of *Saccharomyces cerevisiae* CEN.PK113-1A,” *BMC systems biology*, vol. 2, p. 60, aug 2008.
- [9] O. Fornes, J. A. Castro-Mondragon, A. Khan, R. Van Der Lee, X. Zhang, P. A. Richmond, B. P. Modi, S. Correard, M. Gheorghe, D. Baranašić, W. Santana-Garcia, G. Tan, J. Chèneby, B. Ballester, F. Parcy, A. Sandelin, B. Lenhard, W. W. Wasserman, and A. Mathelier, “JASPAR 2020: Update of the open-Access database of transcription factor binding profiles,” *Nucleic Acids Research*, vol. 48, no. D1, pp. D87–D92, 2020.
- [10] S. B. Needleman and C. D. Wunsch, “A general method applicable to the search for similarities in the amino acid sequence of two proteins,” *Journal of Molecular Biology*, vol. 48, pp. 443–453, mar 1970.
- [11] L. A. Cauchy, “Methode generale pour la resolution des systemes d’equations simultanees,” *C.R. Acad. Sci. Paris*, vol. 25, pp. 536–538, 1847.
- [12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.
- [13] D. P. Kingma and J. L. Ba, “Adam: A Method for Stochastic Optimization,” *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, dec 2014.
- [14] W. H. Brondyk, “Chapter 11 Selecting an Appropriate Method for Expressing a Recombinant Protein,” in *Methods in Enzymology*, vol. 463, pp. 131–147, Academic Press Inc., jan 2009.
- [15] E. Çelik and P. Çalık, “Production of recombinant proteins by yeast cells,” *Biotechnology Advances*, vol. 30, pp. 1108–1118, sep 2012.
- [16] R. L. Strausberg and S. L. Strausberg, “Overview of Protein Expression in *Saccharomyces cerevisiae*,” *Current Protocols in Protein Science*, vol. 2, pp. 5.6.1–5.6.7, may 1995.
- [17] J. M. Cregg, I. Tolstorukov, A. Kusari, J. Sunga, K. Madden, and T. Chappell, “Chapter 13 Expression in the Yeast *Pichia pastoris*,” in *Methods in Enzymology*, vol. 463, pp. 169–189, Academic Press Inc., jan 2009.

- [18] P. Çalık, Ö. Ata, H. Güneş, A. Massahi, E. Boy, A. Keskin, S. Öztürk, G. H. Zerze, and T. H. Özdamar, “Recombinant protein production in *Pichia pastoris* under glyceraldehyde-3-phosphate dehydrogenase promoter: From carbon source metabolism to bioreactor operation parameters,” *Biochemical Engineering Journal*, vol. 95, pp. 20–36, 2015.
- [19] U. Alon, *An introduction to systems biology: Design principles of biological circuits*. 2006.
- [20] Ö. Kalender and P. Çalık, “Transcriptional regulatory proteins in central carbon metabolism of *Pichia pastoris* and *Saccharomyces cerevisiae*,” *Applied Microbiology and Biotechnology*, vol. 104, pp. 7273–7311, sep 2020.
- [21] J. W. Lee, M. S. Han, S. Choi, J. Yi, T. W. Lee, and S. Y. Lee, “Organic Acids: Succinic and Malic Acids,” in *Comprehensive Biotechnology, Second Edition*, vol. 3, pp. 149–161, Elsevier Inc., sep 2011.
- [22] I. Otero-Muras, A. A. Mannan, J. R. Banga, and D. A. Oyarzún, “Multi-objective optimization of gene circuits for metabolic engineering,” in *IFAC-PapersOnLine*, vol. 52, pp. 13–16, Elsevier B.V., jan 2019.
- [23] Y. Zeng, M. Gong, M. Lin, D. Gao, and Y. Zhang, “A Review about Transcription Factor Binding Sites Prediction Based on Deep Learning,” *IEEE Access*, 2020.
- [24] B. G. Ergün, B. Gasser, D. Mattanovich, and P. Çalık, “Engineering of alcohol dehydrogenase 2 hybrid-promoter architectures in *Pichia pastoris* to enhance recombinant protein expression on ethanol,” *Biotechnology and Bioengineering*, vol. 116, pp. 2674–2686, oct 2019.
- [25] B. G. Ergün, Demir, T. H. Özdamar, B. Gasser, D. Mattanovich, and P. Çalık, “Engineered Deregulation of Expression in Yeast with Designed Hybrid-Promoter Architectures in Coordination with Discovered Master Regulator Transcription Factor,” *Advanced Biosystems*, vol. 4, p. 1900172, apr 2020.
- [26] B. G. Ergün and P. Çalık, “Hybrid-architected promoter design to deregulate expression in yeast,” *Methods in enzymology*, vol. 660, pp. 105–125, 2021.

- [27] B. G. Ergün and P. Çalık, “Chapter Four - Hybrid-architected promoter design to engineer expression in yeast,” in *Recombinant Protein Expression: Eukaryotic Hosts* (W. B. O’Dell and Z. Kelman, eds.), vol. 660 of *Methods in Enzymology*, pp. 81–104, Academic Press, 2021.
- [28] A. Goffeau, B. G. Barrell, H. Bussey, R. W. Davis, B. Dujon, H. Feldmann, F. Galibert, J. D. Hoheisel, C. Jacq, M. Johnston, E. J. Louis, H. W. Mewes, Y. Murakami, P. Philippsen, H. Tettelin, and S. G. Oliver, “Life with 6000 genes,” *Science (New York, N.Y.)*, vol. 274, pp. 546,563–567, oct 1996.
- [29] S. G. Oliver, A. Lock, M. A. Harris, P. Nurse, and V. Wood, “Model organism databases: essential resources that need the support of both funders and users,” *BMC Biology*, vol. 14, no. 1, p. 49, 2016.
- [30] D. A. Peña, B. Gasser, J. Zanghellini, M. G. Steiger, and D. Mattanovich, “Metabolic engineering of *Pichia pastoris*,” nov 2018.
- [31] Z. Yang and Z. Zhang, “Engineering strategies for enhanced production of protein and bio-products in *Pichia pastoris*: A review,” *Biotechnology advances*, vol. 36, no. 1, pp. 182–195, 2018.
- [32] M. A. Nieto-Taype, X. Garcia-Ortega, J. Albiol, J. L. Montesinos-Seguí, and F. Valero, “Continuous Cultivation as a Tool Toward the Rational Bioprocess Development With *Pichia Pastoris* Cell Factory,” *Frontiers in bioengineering and biotechnology*, vol. 8, p. 632, 2020.
- [33] D. Dikicioglu, V. Wood, K. M. Rutherford, M. D. McDowall, and S. G. Oliver, “Improving functional annotation for industrial microbes: a case study with *Pichia pastoris*,” *Trends in Biotechnology*, vol. 32, no. 8, pp. 396–399, 2014.
- [34] Z. Wang, W. He, J. Tang, and F. Guo, “Identification of Highest-Affinity Binding Sites of Yeast Transcription Factor Families,” *Journal of Chemical Information and Modeling*, vol. 60, pp. 1876–1883, mar 2020.
- [35] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” 2013.
- [36] D. S. Latchman, “Transcription factors: An overview,” *The International Journal of Biochemistry & Cell Biology*, vol. 29, no. 12, pp. 1305–1312, 1997.

- [37] D. L. Nelson and M. M. Cox, *Lehninger Principles of Biochemistry 5th Ed.* New York: W. H. Freeman and Company, 5 ed., 2008.
- [38] M. J. Solomon, P. L. Larsen, and A. Varshavsky, “Mapping protein-DNA interactions in vivo with formaldehyde: evidence that histone H4 is retained on a highly transcribed gene.,” *Cell*, vol. 53, pp. 937–947, jun 1988.
- [39] E. R. Mardis, “ChIP-seq: welcome to the new frontier,” *Nature Methods*, vol. 4, no. 8, pp. 613–614, 2007.
- [40] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, T. L. Volkert, C. J. Wilson, S. P. Bell, and R. A. Young, “Genome-wide location and function of DNA binding proteins.,” *Science (New York, N.Y.)*, vol. 290, pp. 2306–2309, dec 2000.
- [41] D. R. Bentley, “Whole-genome re-sequencing,” *Current Opinion in Genetics & Development*, vol. 16, no. 6, pp. 545–552, 2006.
- [42] G. Robertson, M. Hirst, M. Bainbridge, M. Bilenky, Y. Zhao, T. Zeng, G. Euskirchen, B. Bernier, R. Varhol, A. Delaney, N. Thiessen, O. L. Griffith, A. He, M. Marra, M. Snyder, and S. Jones, “Genome-wide profiles of STAT1 DNA association using chromatin immunoprecipitation and massively parallel sequencing,” *Nature Methods*, vol. 4, no. 8, pp. 651–657, 2007.
- [43] M. F. Berger and M. L. Bulyk, “Universal protein-binding microarrays for the comprehensive characterization of the DNA-binding specificities of transcription factors,” *Nature Protocols*, vol. 4, no. 3, pp. 393–411, 2009.
- [44] P. T. Monteiro, J. Oliveira, P. Pais, M. Antunes, M. Palma, M. Cavaleiro, M. Galocha, C. P. Godinho, L. C. Martins, N. Bourbon, M. N. Mota, R. A. Ribeiro, R. Viana, I. Sá-Correia, and M. C. Teixeira, “YEASTRACT+: A portal for cross-species comparative genomics of transcription regulation in yeasts,” *Nucleic Acids Research*, vol. 48, no. D1, pp. D642–D649, 2020.
- [45] M. A. Hume, L. A. Barrera, S. S. Gisselbrecht, and M. L. Bulyk, “UniPROBE, update 2015: new tools and content for the online database of protein-

- binding microarray data on protein–DNA interactions,” *Nucleic Acids Research*, vol. 43, pp. D117–D122, jan 2015.
- [46] B. Dujon, “The yeast genome project: what did we learn?,” *Trends in Genetics*, vol. 12, no. 7, pp. 263–270, 1996.
- [47] T. Pfeiffer and A. Morley, “An evolutionary perspective on the Crabtree effect,” *Frontiers in molecular biosciences*, vol. 1, p. 17, 2014.
- [48] C. Zhu, K. J. Byers, R. P. McCord, Z. Shi, M. F. Berger, D. E. Newburger, K. Saulrieta, Z. Smith, M. V. Shah, M. Radhakrishnan, A. A. Philippakis, Y. Hu, F. De Masi, M. Pacek, A. Rolfs, T. Murthy, J. Labaer, and M. L. Bulyk, “High-resolution DNA-binding specificity analysis of yeast transcription factors,” *Genome research*, vol. 19, pp. 556–566, apr 2009.
- [49] K. D. MacIsaac, T. Wang, D. B. Gordon, D. K. Gifford, G. D. Stormo, and E. Fraenkel, “An improved map of conserved regulatory sites for *Saccharomyces cerevisiae*,” *BMC Bioinformatics*, vol. 7, pp. 1–14, mar 2006.
- [50] J. Heringa, *Post-genome Informatics*. Oxford University Press, jul 2000.
- [51] A. Bateman, M. J. Martin, S. Orchard, M. Magrane, R. Agivetova, S. Ahmad, E. Alpi, E. H. Bowler-Barnett, R. Britto, B. Bursteinas, H. Bye-A-Jee, R. Coetzee, A. Cukura, A. D. Silva, P. Denny, T. Dogan, T. G. Ebenezer, J. Fan, L. G. Castro, P. Garmiri, G. Georgiou, L. Gonzales, E. Hatton-Ellis, A. Hussein, A. Ignatchenko, G. Insana, R. Ishtiaq, P. Jokinen, V. Joshi, D. Jyothi, A. Lock, R. Lopez, A. Luciani, J. Luo, Y. Lussi, A. MacDougall, F. Madeira, M. Mahmoudy, M. Menchi, A. Mishra, K. Moulang, A. Nightingale, C. S. Oliveira, S. Pundir, G. Qi, S. Raj, D. Rice, M. R. Lopez, R. Saidi, J. Sampson, T. Sawford, E. Speretta, E. Turner, N. Tyagi, P. Vasudev, V. Volynkin, K. Warner, X. Watkins, R. Zaru, H. Zellner, A. Bridge, S. Poux, N. Redaschi, L. Aimo, G. Argoud-Puy, A. Auchincloss, K. Axelsen, P. Bansal, D. Baratin, M. C. Blatter, J. Bolleman, E. Boutet, L. Breuza, C. Casals-Casas, E. de Castro, K. C. Echioukh, E. Coudert, B. Cuhe, M. Doche, D. Dornevil, A. Estreicher, M. L. Famiglietti, M. Feuermann, E. Gasteiger, S. Gehant, V. Gerritsen, A. Gos, N. Gruaz-Gumowski, U. Hinz, C. Hulo, N. Hyka-Nouspikel, F. Jungo,

G. Keller, A. Kerhornou, V. Lara, P. Le Mercier, D. Lieberherr, T. Lombardot, X. Martin, P. Masson, A. Morgat, T. B. Neto, S. Paesano, I. Pedruzzi, S. Pilbout, L. Pourcel, M. Pozzato, M. Pruess, C. Rivoire, C. Sigrist, K. Sonesson, A. Stutz, S. Sundaram, M. Tognolli, L. Verbregue, C. H. Wu, C. N. Arighi, L. Arminski, C. Chen, Y. Chen, J. S. Garavelli, H. Huang, K. Laiho, P. McGarvey, D. A. Natale, K. Ross, C. R. Vinayaka, Q. Wang, Y. Wang, L. S. Yeh, and J. Zhang, “UniProt: the universal protein knowledgebase in 2021,” *Nucleic Acids Research*, vol. 49, pp. D480–D489, jan 2021.

- [52] E. W. Sayers, E. E. Bolton, J. R. Brister, K. Canese, J. Chan, D. C. Comeau, R. Connor, K. Funk, C. Kelly, S. Kim, T. Madej, A. Marchler-Bauer, C. Lanczycki, S. Lathrop, Z. Lu, F. Thibaud-Nissen, T. Murphy, L. Phan, Y. Skripchenko, T. Tse, J. Wang, R. Williams, B. W. Trawick, K. D. Pruitt, and S. T. Sherry, “Database resources of the national center for biotechnology information.,” *Nucleic acids research*, vol. 50, pp. D20–D26, jan 2022.
- [53] G. P. Cereghino, J. L. Cereghino, C. Ilgen, and J. M. Cregg, “Production of recombinant proteins in fermenter cultures of the yeast *Pichia pastoris*,” *Current Opinion in Biotechnology*, vol. 13, no. 4, pp. 329–332, 2002.
- [54] D. Mattanovich, A. Graf, J. Stadlmann, M. Dragosits, A. Redl, M. Maurer, M. Kleinheinz, M. Sauer, F. Altmann, and B. Gasser, “Genome, secretome and glucose transport highlight unique features of the protein production host *Pichia pastoris*,” *Microbial cell factories*, vol. 8, p. 29, jun 2009.
- [55] P. Katara, A. Grover, and V. Sharma, “Phylogenetic footprinting: a boost for microbial regulatory genomics,” *Protoplasma*, vol. 249, no. 4, pp. 901–907, 2012.
- [56] G. D. Stormo, “DNA binding sites: representation and discovery,” *Bioinformatics*, vol. 16, pp. 16–23, jan 2000.
- [57] W. W. Wasserman and A. Sandelin, “Applied bioinformatics for the identification of regulatory elements,” *Nature Reviews Genetics*, vol. 5, no. 4, pp. 276–287, 2004.

- [58] J. M. Heumann, A. S. Lapedes, and G. D. Stormo, "Neural networks for determining protein specificity and multiple alignment of binding sites," in *Intelligent Systems for Molecular Biology*, pp. 188–194, 1994.
- [59] T. D. Schneider and R. M. Stephens, "Sequence logos: A new way to display consensus sequences," *Nucleic Acids Research*, vol. 18, pp. 6097–6100, oct 1990.
- [60] T. D. Schneider, G. D. Stormo, L. Gold, and A. Ehrenfeucht, "Information content of binding sites on nucleotide sequences," *Journal of Molecular Biology*, vol. 188, pp. 415–431, apr 1986.
- [61] L. Pickert, I. Reuter, F. Klawonn, and E. Wingender, "Transcription regulatory region analysis using signal detection and fuzzy clustering.," *Bioinformatics*, vol. 14, pp. 244–251, jan 1998.
- [62] K. Cartharius, K. Frech, K. Grote, B. Klocke, M. Haltmeier, A. Klingenhoff, M. Frisch, M. Bayerlein, and T. Werner, "MatInspector and beyond: promoter analysis based on transcription factor binding sites," *Bioinformatics*, vol. 21, pp. 2933–2942, jul 2005.
- [63] E. Berezikov, V. Guryev, R. H. Plasterk, and E. Cuppen, "CONREAL: Conserved Regulatory Elements Anchored Alignment Algorithm for Identification of Transcription Factor Binding Sites by Phylogenetic Footprinting," *Genome Research*, vol. 14, p. 170, jan 2004.
- [64] D. A. Tagle, B. F. Koop, M. Goodman, J. L. Slightom, D. L. Hess, and R. T. Jones, "Embryonic ϵ and γ globin genes of a prosimian primate (*Galago crassicaudatus*): Nucleotide and amino acid sequences, developmental regulation and phylogenetic footprints," *Journal of Molecular Biology*, vol. 203, pp. 439–455, sep 1988.
- [65] M. Blanchette and M. Tompa, "Discovery of Regulatory Elements by a Computational Method for Phylogenetic Footprinting," *Genome Research*, vol. 12, no. 5, p. 739, 2002.
- [66] S. Nepf and M. Tompa, "MicroFootPrinter: a tool for phylogenetic footprint-

ing in prokaryotic genomes,” *Nucleic Acids Research*, vol. 34, pp. W366–W368, jul 2006.

- [67] L. Duret and P. Bucher, “Searching for regulatory elements in human noncoding sequences,” *Current Opinion in Structural Biology*, vol. 7, pp. 399–406, jun 1997.
- [68] L. Bernauer, A. Radkohl, L. G. K. Lehmayr, and A. Emmerstorfer-Augustin, “*Komagataella phaffii* as Emerging Model Organism in Fundamental Research,” *Frontiers in Microbiology*, vol. 11, p. 3462, jan 2021.
- [69] M. Blanchette, B. Schwikowski, and M. Tompa, “Algorithms for phylogenetic footprinting,” *Journal of computational biology : a journal of computational molecular cell biology*, vol. 9, no. 2, pp. 211–223, 2002.
- [70] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” *Journal of Molecular Biology*, vol. 215, pp. 403–410, oct 1990.
- [71] W. R. Pearson, “Empirical statistical estimates for sequence similarity searches,” *Journal of Molecular Biology*, vol. 276, pp. 71–84, feb 1998.
- [72] M. Tuğrul, T. Paixão, N. H. Barton, and G. Tkačik, “Dynamics of Transcription Factor Binding Site Evolution,” *PLOS Genetics*, vol. 11, p. e1005639, nov 2015.
- [73] T. K. Man and G. D. Stormo, “Non-independence of Mnt repressor–operator interaction determined by a new quantitative multiple fluorescence relative affinity (QuMFRA) assay,” *Nucleic Acids Research*, vol. 29, pp. 2471–2478, jun 2001.
- [74] M. L. Bulyk, P. L. Johnson, and G. M. Church, “Nucleotides of transcription factor binding sites exert interdependent effects on the binding affinities of transcription factors,” *Nucleic acids research*, vol. 30, pp. 1255–1261, mar 2002.
- [75] S. J. Maerkl and S. R. Quake, “A systems approach to measuring the binding energy landscapes of transcription factors,” *Science*, vol. 315, pp. 233–237, jan 2007.

- [76] R. Siddharthan, “Dinucleotide Weight Matrices for Predicting Transcription Factor Binding Sites: Generalizing the Position Weight Matrix,” *PLOS ONE*, vol. 5, no. 3, p. e9722, 2010.
- [77] A. Mathelier and W. W. Wasserman, “The Next Generation of Transcription Factor Binding Site Prediction,” *PLOS Computational Biology*, vol. 9, p. e1003214, sep 2013.
- [78] M. Santolini, T. Mora, and V. Hakim, “A General Pairwise Interaction Model Provides an Accurate Description of In Vivo Transcription Factor Binding Sites,” *PLOS ONE*, vol. 9, p. e99015, jun 2014.
- [79] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 3 ed., 2010.
- [80] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [81] L. Breiman, “Random Forests,” *Machine Learning 2001 45:1*, vol. 45, pp. 5–32, oct 2001.
- [82] O. C. Ibe, “Functions of Random Variables,” in *Fundamentals of Applied Probability and Random Processes*, pp. 185–223, Academic Press, jan 2014.
- [83] M. Mohri, A. Rostamizadeh, and A. Talwalkar, *Foundations of Machine Learning*, vol. 60. MITPress, 2 ed., 2018.
- [84] S. Nembrini, I. R. König, and M. N. Wright, “The revival of the Gini importance?,” *Bioinformatics*, vol. 34, p. 3711, nov 2018.
- [85] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees*. New York: Chapman & Hall, jan 1984.
- [86] H. Ishwaran, “The effect of splitting on random forests,” *Machine Learning*, vol. 99, pp. 75–118, apr 2015.
- [87] B. Hooghe, S. Broos, F. Van Roy, and P. De Bleser, “A flexible integrative approach based on random forest improves prediction of transcription factor binding sites,” *Nucleic acids research*, vol. 40, aug 2012.

- [88] C. S. Smitha and R. Saritha, “Computational transcription factor binding prediction using random forests,” *2014 International Conference on Control, Instrumentation, Communication and Computational Technologies, ICCICCT 2014*, pp. 577–583, dec 2014.
- [89] A. A. Antikainen, M. Heinonen, and H. Lähdesmäki, “Modeling binding specificities of transcription factor pairs with random forests,” *BMC Bioinformatics* 2022 23:1, vol. 23, pp. 1–17, jun 2022.
- [90] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-Aug, pp. 785–794, 2016.
- [91] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors),” *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.
- [92] J. H. Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis*, vol. 38, pp. 367–378, feb 2002.
- [93] J. H. Friedman, “Greedy function approximation: A gradient boosting machine.,” *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.
- [94] P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. PhD thesis, Harvard University, 1974.
- [95] S. Linnainmaa, “Taylor expansion of the accumulated rounding error,” *BIT Numerical Mathematics* 1976 16:2, vol. 16, no. 2, pp. 146–160, 1976.
- [96] Y. Lecun, “Une procedure d’apprentissage pour reseau a seuil asymmetrique (A learning scheme for asymmetric threshold networks),” in *Proceedings of Cognitiva 85, Paris, France*, pp. 599–604, 1985.
- [97] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning Internal Representations by Error Propagation,” in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations* (D. E. Rumelhart and J. L. McClelland, eds.), pp. 318–362, Cambridge, MA: MIT Press, 1986.

- [98] G. E. Hinton, “What Kind of a Graphical Model is the Brain?,” in *Proceedings of the 19th International Joint Conference on Artificial Intelligence, IJCAI’05*, (San Francisco, CA, USA), pp. 1765–1775, Morgan Kaufmann Publishers Inc., 2005.
- [99] G. E. Hinton and R. R. Salakhutdinov, “Reducing the Dimensionality of Data with Neural Networks,” *Science*, vol. 313, pp. 504–507, jul 2006.
- [100] G. E. Hinton, S. Osindero, and Y.-W. Teh, “A fast learning algorithm for deep belief nets,” *Neural computation*, vol. 18, pp. 1527–1554, jul 2006.
- [101] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, “Greedy Layer-Wise Training of Deep Networks,” in *Proceedings of the 19th International Conference on Neural Information Processing Systems, NIPS’06*, (Cambridge, MA, USA), pp. 153–160, MIT Press, 2006.
- [102] B. Schölkopf, J. Platt, and T. Hofmann, *Efficient Learning of Sparse Representations with an Energy-Based Model*, pp. 1137–1144. 2007.
- [103] P. Sermanet, K. Kavukcuoglu, S. Chintala, and Y. LeCun, “Pedestrian Detection with Unsupervised Multi-Stage Feature Learning,” *CoRR*, vol. abs/1212.0, 2012.
- [104] Y. Lecun, *Generalization and network design strategies*. Elsevier, 1989.
- [105] B. T. Polyak, “Some methods of speeding up the convergence of iteration methods,” *USSR Computational Mathematics and Mathematical Physics*, vol. 4, pp. 1–17, jan 1964.
- [106] R. Zaheer and H. Shaziya, “A Study of the Optimization Algorithms in Deep Learning,” in *2019 Third International Conference on Inventive Systems and Control (ICISC)*, pp. 536–539, 2019.
- [107] J. Brownlee, *Probability for Machine Learning: Discover How To Harness Uncertainty With Python*. Machine Learning Mastery, 2019.
- [108] K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun, “What is the best multi-stage architecture for object recognition?,” in *2009 IEEE 12th International Conference on Computer Vision*, pp. 2146–2153, 2009.

- [109] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” in *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML’10, (Madison, WI, USA), pp. 807–814, Omnipress, 2010.
- [110] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks,” in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics* (G. Gordon, D. Dunson, and M. Dudík, eds.), vol. 15 of *Proceedings of Machine Learning Research*, (Fort Lauderdale, FL, USA), pp. 315–323, PMLR, 2011.
- [111] A. L. Maas, A. Y. Hannun, and A. Y. Ng, “Rectifier nonlinearities improve neural network acoustic models,” in *ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013.
- [112] D. A. Clevert, T. Unterthiner, and S. Hochreiter, “Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs),” in *4th International Conference on Learning Representations, ICLR 2016 - Conference Track Proceedings*, International Conference on Learning Representations, ICLR, nov 2015.
- [113] F. Mordelet, J. Horton, A. J. Hartemink, B. E. Engelhardt, and R. Gordâ, “Stability selection for regression-based models of transcription factor-DNA binding specificity,” *Bioinformatics*, vol. 29, pp. 117–125, 2013.
- [114] Y. Z. Chen, Y. R. Tang, Z. Y. Sheng, and Z. Zhang, “Prediction of mucin-type O-glycosylation sites in mammalian proteins using the composition of k-spaced amino acid pairs,” *BMC bioinformatics*, vol. 9, p. 101, feb 2008.
- [115] X. Wang, R. Yan, and J. Song, “DephosSite: a machine learning approach for discovering phosphatase-specific dephosphorylation sites,” *Scientific Reports 2016 6:1*, vol. 6, pp. 1–11, mar 2016.
- [116] X. Wang and R. Yan, “RFathM6A: a new tool for predicting m⁶A sites in *Arabidopsis thaliana*,” *Plant Molecular Biology*, vol. 96, pp. 327–337, 2018.
- [117] W. Chen, H. Yang, P. Feng, H. Ding, and H. Lin, “iDNA4mC: identifying DNA N 4-methylcytosine sites based on nucleotide chemical properties,” *Bioinformatics*, 2017.

- [118] G. Pan, Id, L. Jiang, J. Tang, and F. Guo, “A Novel Computational Method for Detecting DNA Methylation Sites with DNA Sequence Information and Physicochemical Properties,” *International Journal of Molecular Sciences Article*, 2018.
- [119] W. He, C. Jia, and Q. Zou, “4mCPred: machine learning methods for DNA N4-methylcytosine sites prediction,” *Bioinformatics*, vol. 35, pp. 593–601, feb 2019.
- [120] W. He and C. Jia, “EnhancerPred2.0: predicting enhancers and their strength based on position-specific trinucleotide propensity and electron–ion interaction potential feature selection,” *Molecular BioSystems*, vol. 13, pp. 767–774, mar 2017.
- [121] C. Jia, Q. Yang, and Q. Zou, “NucPosPred: Predicting species-specific genomic nucleosome positioning via four different modes of general PseKNC,” *Journal of theoretical biology*, vol. 450, pp. 15–21, aug 2018.
- [122] S. S. Sahu and G. Panda, “Efficient localization of hot spots in proteins using a novel S-transform based filtering approach,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 8, no. 5, pp. 1235–1246, 2011.
- [123] Z. Ur-Rehman and A. Khan, “G-protein-coupled receptor prediction using pseudo-amino-acid composition and multiscale energy representation of different physiochemical properties,” *Analytical Biochemistry*, vol. 412, pp. 173–182, may 2011.
- [124] M. Johnson, I. Zaretskaya, Y. Raytselis, Y. Merezuk, S. McGinnis, and T. L. Madden, “NCBI BLAST: a better web interface.,” *Nucleic acids research*, vol. 36, no. Web Server issue, pp. 5–9, 2008.
- [125] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski, and M. J. De Hoon, “Biopython: Freely available Python tools for computational molecular biology and bioinformatics,” *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009.
- [126] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos,

- D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [127] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, X. Zheng, and G. Research, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,” oct 2015.
- [128] G. Hughes, “On the mean accuracy of statistical pattern recognizers,” *IEEE Transactions on Information Theory*, vol. 14, no. 1, pp. 55–63, 1968.
- [129] R. E. Bellman, *Dynamic Programming*. Rand Corporation research study, Princeton University Press, 1957.



APPENDIX A

PROTEIN-PROTEIN BLAST RESULTS

Table A.1: Protein-protein BLASTs of the glycolysis enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Coverage (%)	E-value	Identity (%)
HXK2 852639	PAS_chr1-4_0561 8197692	543	95	0	58
PGI1 852495	PAS_chr3_0456 8199584	888	98	0	77
PFK1 853155	PAS_chr2-1_0402 8198870	1129	98	0	57
PFK2 855245	PAS_chr1-4_0047 8196884	1130	98	0	58
FBA1 853805	PAS_chr1-1_0072 8197200	590	98	0	78
TPI1 851620	PAS_chr3_0951 8200302	366	100	4e-130	72
TDH2 853465	PAS_chr2-1_0437 8198905	557	99	0	80
TDH3 853106	PAS_chr2-1_0437 8198905	560	99	0	81
Continued on next page					

Table A.1 – continued from previous page

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Covery (%)	E-value	Identity (%)
PGK1 850370	PAS_chr1-4_0292 8197742	635	99	0	75
GPM1 853705	PAS_chr3_0826 8200319	395	100	1e-141	78
ENO2 856579	PAS_chr3_0082 8199366	685	98	0	78
CDC19 851193	PAS_chr2-1_0769 8198046	758	99	0	72

Table A.2: Protein-protein BLASTs of the gluconeogenesis enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Covery (%)	E-value	Identity (%)
PCK1 853972	PAS_FragB_0061 8197501	827	99	0	69
ENO1 853169	PAS_chr3_0082 8199366	694	98	0	80
GPM1 853705	PAS_chr3_0826 8200319	395	100	1e-141	78
PGK1 850370	PAS_chr1-4_0292 8197742	635	99	0	75
TDH2 853465	PAS_chr2-1_0437 8198905	557	99	0	80
TDH3 853106	PAS_chr2-1_0437 8198905	560	99	0	81
Continued on next page					

Table A.2 – continued from previous page

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Coverage (%)	E-value	Identity (%)
FBA1 853805	PAS_chr1-1_0072 8197200	590	98	0	78
FBP1 851092	PAS_chr3_0868 8199670	484	95	2e-173	67
PGI1 852495	PAS_chr3_0456 8199584	888	98	0	77

Table A.3: Protein-protein BLASTs of the Pentose Phosphate Pathway enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Coverage (%)	E-value	Identity (%)
SOL3 856568	PAS_chr3_1126 8200158	210	99	1e-68	43
SOL4 853163	PAS_chr3_1126 8200158	168	96	3e-52	36
GND1 856589	PAS_chr3_0277 8200105	796	99	0	79
RKI1 854262	PAS_chr4_0213 8200884	290	93	5e-100	59
RPE1 853322	AT250_GQ6803479 -	347	100	3e-123	70
TKL1 856188	PAS_chr1-4_0150 8197134	952	98	0	69
TAL1 851068	PAS_chr2-2_0337 8198237	487	96	3e-175	73

Table A.4: Protein-protein BLASTs of the Alcoholic Fermentation enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Coverage (%)	E-value	Identity (%)
PDC1 850733	PAS_chr3_0188 8200158	754	99	0	64
ADH1 854068	PAS_chr2-1_0472 8200158	497	99	1e-178	73

Table A.5: Protein-protein BLASTs of the TCA Cycle enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Coverage (%)	E-value	Identity (%)
PDB1 852522	PAS_chr1-4_0593 8197723	543	89	0	76
LAT1 855653	PAS_chr1-1_0050 8197567	534	100	0	59
LPD1 850527	PAS_chr2-2_0048 8199153	614	97	0	68
PDX1 853107	PAS_chr1-4_0254 8196927	228	99	1e-71	39
CIT1 855732	PAS_chr1-1_0475 8197246	748	99	0	76
ACO1 851013	PAS_chr1-3_0104 8197413	1323	99	0	81
IDH1 855691	PAS_chr4_0580 8200845	468	98	9e-167	64
Continued on next page					

Table A.5 – continued from previous page

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Covary (%)	E-value	Identity (%)
IDH2 854303	PAS_chr2-1_0120 8198516	513	98	0	68
KGD1 854681	PAS_chr2-1_0089 8198485	1492	98	0	70
KGD2 851726	PAS_chr1-3_0094 8196826	496	86	5e-175	64
LSC1 854310	PAS_chr3_0831 8200321	330	92	9e-114	63
LSC2 853159	PAS_chr2-2_0407 8199113	518	93	0	60
SDH1 853709	PAS_chr4_0733 8200628	1031	100	0	78
SDH2 850685	PAS_chr3_1111 8200068	444	99	2e-160	78
SDH3 853716	PAS_chr1-4_0487 8197074	127	75	4e-38	47
SDH4 851758	PAS_chr2-2_0283 8198288	138	71	7e-43	52
FUM1 855866	PAS_chr3_0647 8199846	747	100	0	74
MDH1 853777	PAS_chr2-1_0238 8198787	475	99	1e-170	69

Table A.6: Protein-protein BLASTs of the Glyoxylate Cycle enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Coverage (%)	E-value	Identity (%)
CIT2 850361	PAS_chr1_0475 8197246	684	95	0	72
ACO1 851013	PAS_chr1-3_0104 8197413	1323	99	0	81
ICL1 856794	PAS_chr1-4_0338 8197787	763	98	0	66
MLS1 855606	PAS_chr4_0191 8201098	704	99	0	63
MDH2 853994	PAS_chr4_0815 8201217	260	94	4e-85	44

Table A.7: Protein-protein BLASTs of the Ethanol Utilization Pathway enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol Gene ID	Gene Symbol Gene ID	Score	Coverage (%)	E-value	Identity (%)
ADH2 858349	PAS_chr2-1_0472 8200158	503	99	0	74
ALD4 854556	PAS_chr4_0043 8200158	653	93	0	61
ALD6 856044	PAS_chr3_0987 8200105	542	98	0	53
ACS1 851245	PAS_chr2-1_0767 8200884	890	93	0	67
Continued on next page					

Table A.7 – continued from previous page

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol	Gene Symbol	Score	Cover (%)	E-value	Identity (%)
Gene ID	Gene ID				
ACS2	PAS_chr3_0403	1009	89	0	69
850846	-				

Table A.8: Protein-protein BLASTs of the Glycerol Utilization Pathway enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol	Gene Symbol	Score	Cover (%)	E-value	Identity (%)
Gene ID	Gene ID				
GUT1	PAS_chr4_0783	660	92	0	53
856353	8200561				
GUT2	PAS_chr3_0579	642	98	0	50
854651	8199784				

Table A.9: Protein-protein BLASTs of the Anaplerotic reaction enzymes of *S. cerevisiae* S288c against non-redundant proteins database of *P. pastoris* GS115

<i>S. cerevisiae</i>	<i>P. pastoris</i>	<i>Protein BLAST</i>			
Gene Symbol	Gene Symbol	Score	Cover (%)	E-value	Identity (%)
Gene ID	Gene ID				
MAE1	PAS_chr3_0181	837	89	0	68
853839	8199455				
PYC1	PAS_chr2-2_0024	1838	98	0	77
852818	8198982				
PYC2	PAS_chr2-2_0024	1843	98	0	77
852519	8198982				



APPENDIX B

TFBS SEQUENCE LOGOS FOR *S. CEREVISIAE* AND *P. PASTORIS* TFS

Table B.1: WebLogo Sequence Logos [5] of *S. cerevisiae* S288c PWMs that are extracted from TransFac [6], and computed *P. pastoris* GS115 PWMs

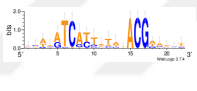
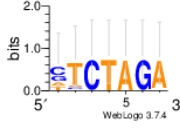
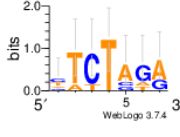
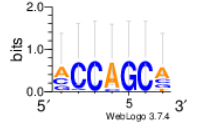
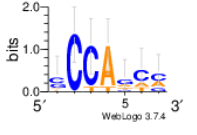
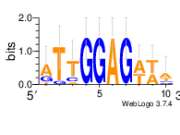
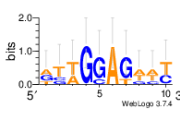
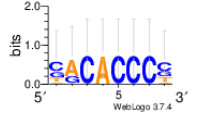
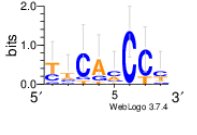
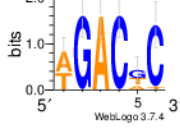
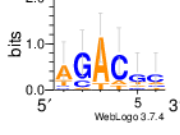
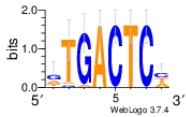
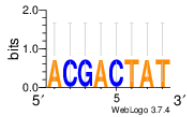
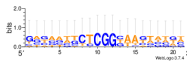
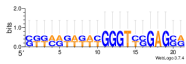
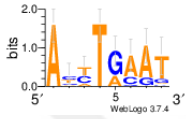
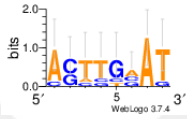
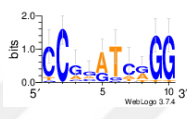
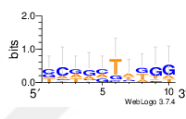
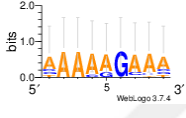
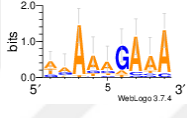
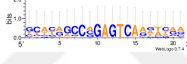
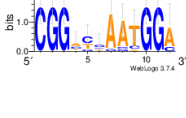
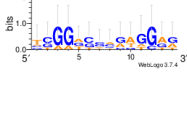
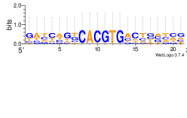
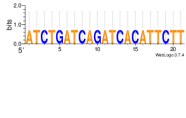
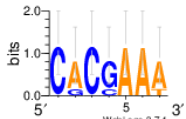
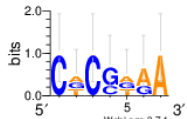
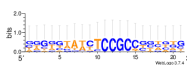
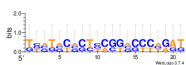
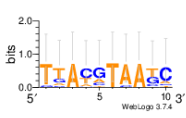
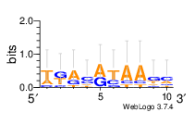
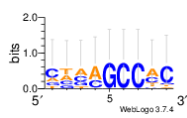
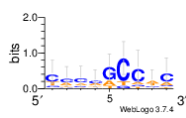
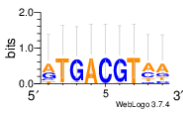
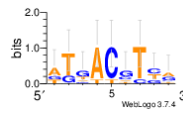
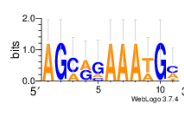
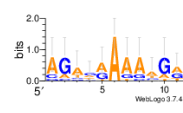
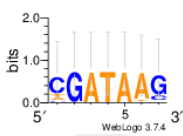
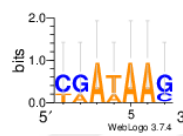
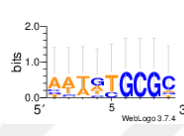
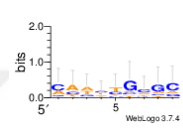
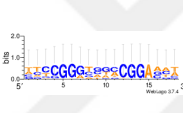
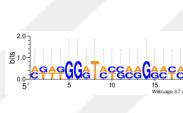
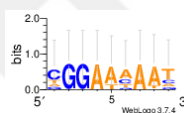
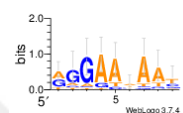
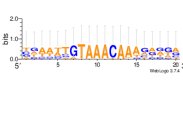
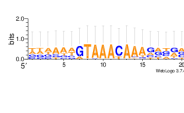
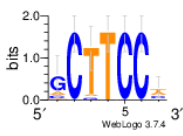
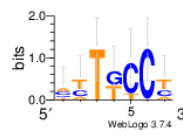
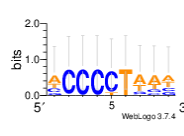
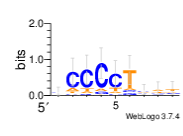
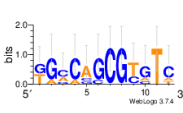
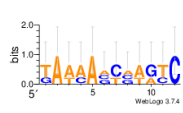
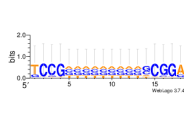
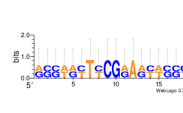
TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Abf1			Abf2		
Ace2			Adr1		
Aft2			Argri		
Continued on next page					

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Argrii			Aro80		
Arr1			Ash1		
Azf1			Bas1		
Cat8			Cbf1		
Ccbf			Cha4		
Cin5			Crz1		

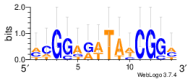
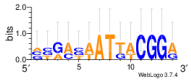
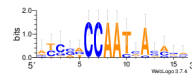
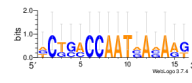
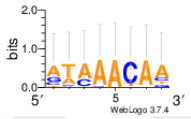
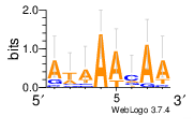
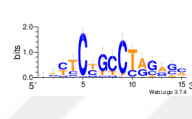
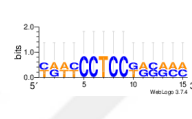
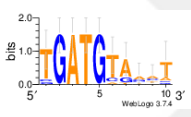
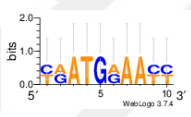
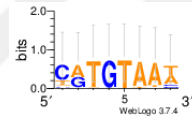
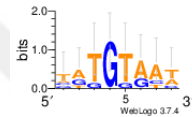
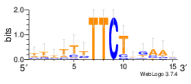
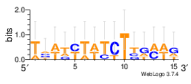
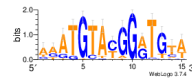
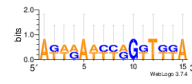
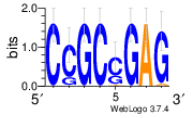
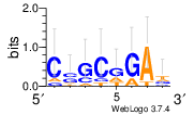
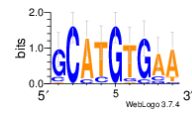
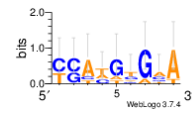
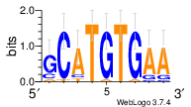
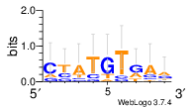
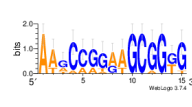
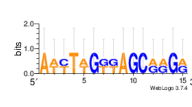
Continued on next page

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Cst6			Cup2		
Dal80			Dal82		
Ecm22			Eds1		
Fkh1			Fkh2		
Ger2			Gis1		
Hac1			Hal9		

Continued on next page

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Hap1			Hap234		
Hcm1			Hmo1		
Hmr1			Hmr2		
Hsf			Ihf1		
Ime1			Ino2		
Ino4			Ixr1		

Continued on next page

Table B.1 – continued from previous page

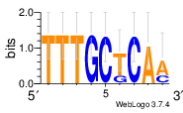
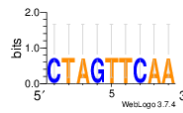
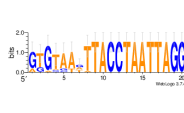
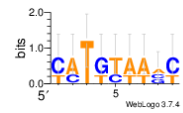
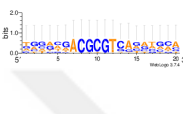
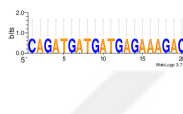
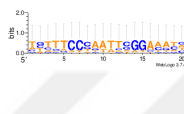
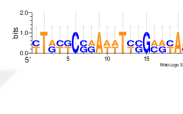
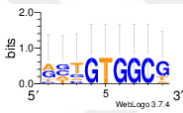
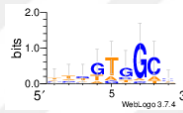
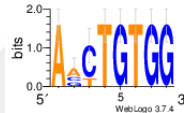
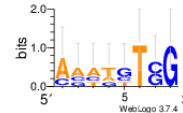
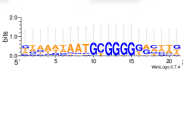
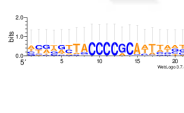
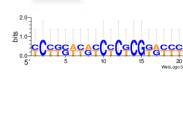
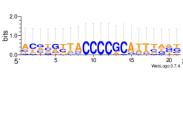
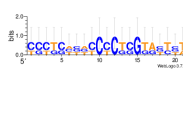
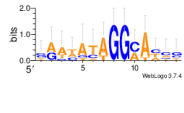
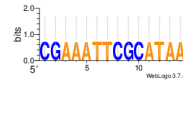
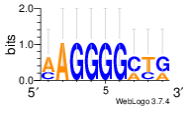
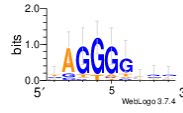
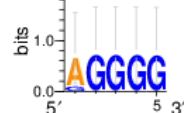
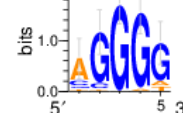
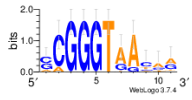
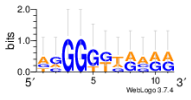
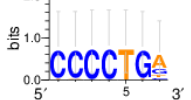
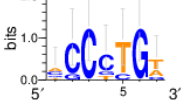
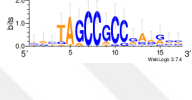
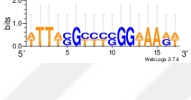
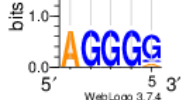
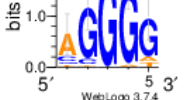
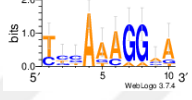
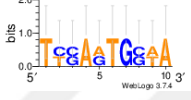
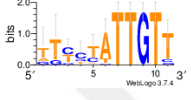
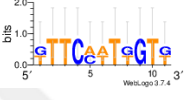
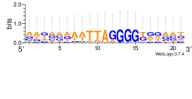
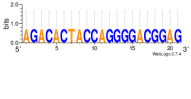
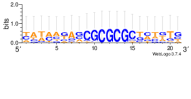
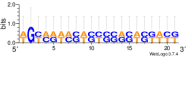
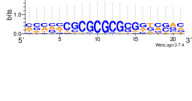
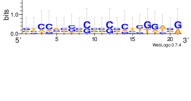
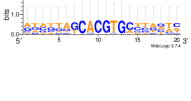
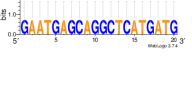
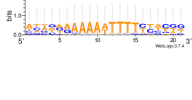
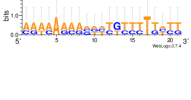
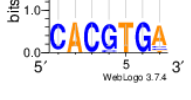
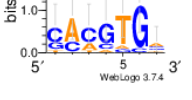
TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Mac1			Matalpha2		
Mbp1			Mcm1		
Met31			Met4		
Mig1			Mig2		
Mig3			Mot3		
Msn2			Msn4		
Continued on next page					

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Nhp10			Nhp6a		
Nrg1			Opi1		
Pdr1			Pdr3		
Pdr8			Phd1		
Put3			Rap1		
Rcs1			Rdr1		

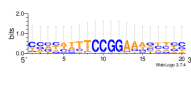
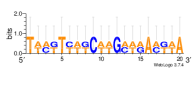
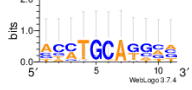
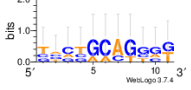
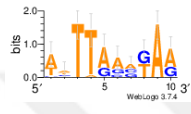
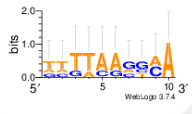
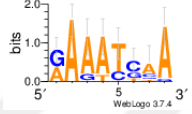
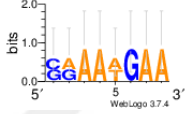
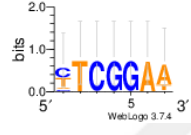
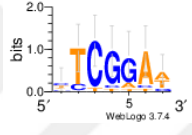
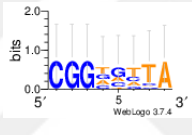
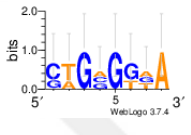
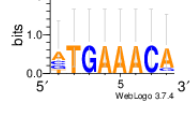
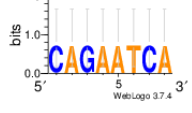
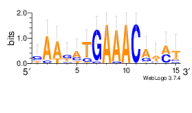
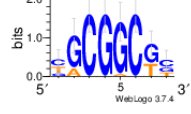
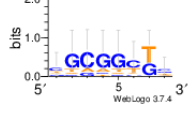
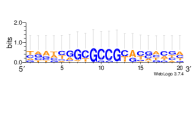
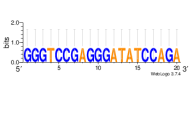
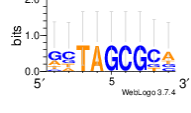
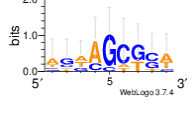
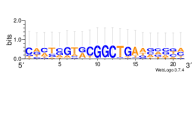
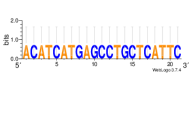
Continued on next page

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Reb1			Rei1		
Repcar1			Rgm1		
Rme1			Rox1		
Rph1			Rsc3		
Rsc30			Rtg3		
Sfp1			Sgc1		

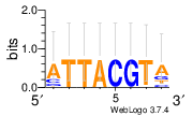
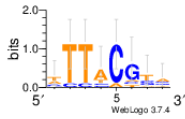
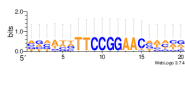
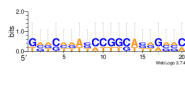
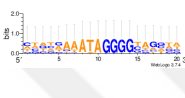
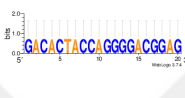
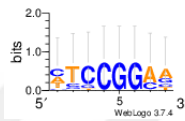
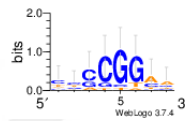
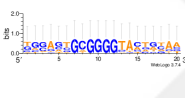
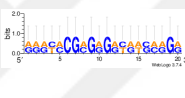
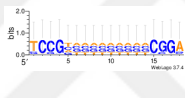
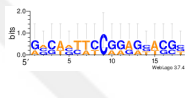
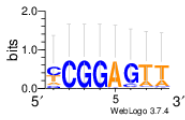
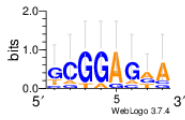
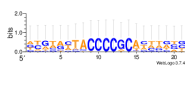
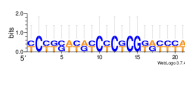
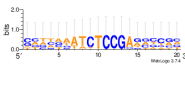
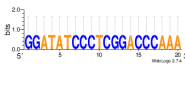
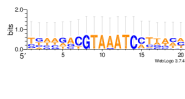
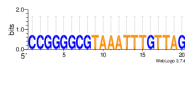
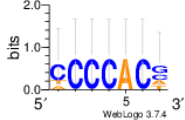
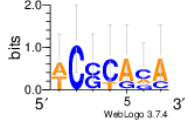
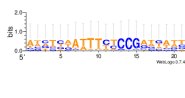
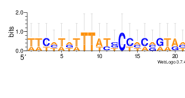
Continued on next page

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Sip4			Sok2		
Spt2			Spt23		
Stb4			Stb5		
Ste12			Ste12p		
Stp1			Stp2		
Stp3			Stp4		

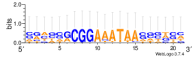
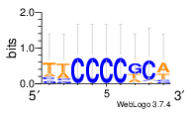
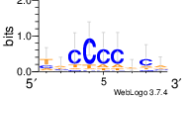
Continued on next page

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Yap3			Ybr239c		
Yer130c			Yer184c		
Ygr067c			Yli054c		
Ylr278c			Yml081w		
Ynr063w			Ypr015c		
Ypr022c			Ypr196w		

Continued on next page

Table B.1 – continued from previous page

TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>	TF	<i>S. cerevisiae</i>	<i>P. pastoris</i>
Yrr1			Zms1		



APPENDIX C

ANNOTATED TFBSS ON *S. CEREVISIAE* AND *P. PASTORIS* PROMOTERS

For all tables ° indicate expression evidence in literature, while * indicates binding evidence in literature. Color codings indicate enzymes pathway.

Table C.1: Number of TFBSSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on *S. cerevisiae* and *P. pastoris* promoters that regulate expression of enzymes that act on Glycolysis and Gluconeogenesis Pathways.

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
HXK2 PAS_chr1-4_0561	Cat8(1), Sip4(3), Ert1(4), Mig1(2) ^{°*}	Adr1(3), Cat8(4), Ert1(1), Stb5(3), Msn2(7), Msn4(2), Mig1(2)	—
PGI1 PAS_chapter3_0456	Sip4(1) [°] , Hap2/3/4/5(1), Stb5(2) [°] , Mig1(1), Gcr1(2) ^{°*}	Cat8(2), Ert1(2), Stb5(1), Msn2(3), Msn4(2), Mig1(1)	—
Continued on next page			

Table C.1 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
PFK1 PAS_chr2-1_0402	Cat8(1), Sip4(3), Ert1(1), Gcr1(2) [°]	Ert1(4), Stb5(1), Msn2(1), Mig1(1)	—
PFK2 PAS_chr1-4_0047	Cat8(1), Sip4(4), Hap2/3/4/5(1), Ert1(2), Stb5(2), Msn2(2) ^{°*} , Msn4(2) [*] , Mig1(1), Gcr1(3) [°]	Stb5(1), Msn2(1), Msn4(1), Mig1(1)	Msn4(1)
FBP1 PAS_chr3_0868	Cat8(3) ^{°*} , Sip4(4) ^{°*} , Hap2/3/4/5(1) [°] , Rds2(1) [*] , Ert1(3) [*] , Stb5(1), Msn2(3) [*] , Msn4(3) [*] , Mig1(1) ^{°*}	Cat8(1), Sip4(2), Stb5(1), Msn2(3), Msn4(1), Mig1(3)	Cat8(1), Msn2(1), Msn4(1)
FBA1 PAS_chr1-1_0072	Adr1(1) ^{°*} , Sip4(2), Rds2(1), Stb5(3), Gcr1(3) [°]	Cat8(2), Ert1(2), Stb5(5), Msn2(8), Msn4(6)	Adr1(1)
Continued on next page			

Table C.1 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
TPH1 PAS_chr3_0951	Cat8(1), Sip4(1), Hap2/3/4/5(1), Msn2(1)*, Msn4(1), Gcr1(2) ^{o*}	Adr1(1), Cat8(3), Stb5(3), Msn2(7), Msn4(3), Mig1(3)	Cat8 (1)
TDH2 PAS_chr2-1_0437	Hap2/3/4/5(1), Rds2(1) ^o , Msn2(2) ^{o*} , Msn4(2)*, Mig1(2), Gcr1(3) ^o	Cat8(2), Hap2/3/4/5(1), Ert1(3), Msn2(3), Msn4(2), Mig1(4)	Hap2/3/4/5(1)
TDH3 PAS_chr2-1_0437	Hap2/3/4/5(1), Msn2(3)*, Msn4(3)*, Mig1(3) ^o , Gcr1(2) ^{o*}		
PGK1 PAS_chr1-4_0292	Cat8(1) ^o , Sip4(2), Msn2(1) ^{o*} , Msn4(1)*, Gcr1(2) ^{o*}	Adr1(1), Cat8(2), Ert1(4), Msn2(1)	—
GPM1 PAS_chr3_0826	Adr1(1), Gcr1(2) ^{o*}	Adr1(1), Cat8(2), Ert1(4), Stb5(1), Msn2(3), Msn4(3)	Adr1(1)
Continued on next page			

Table C.1 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
ENO2 PAS_chr3_0082	Sip4(1), Stb5(1) [°] , Mig1(2)*, Gcr1(2) ^{°*}	Cat8(3), Sip4(2), Ert1(1), Stb5(2), Msn2(6), Msn4(4), Mig1(2)	Msn2(1), Msn4(1), Mig1(1)
ENO1 PAS_chr3_0082	Sip4(1), Stb5(1) [°] , Mig1(2)*, Gcr1(2) ^{°*}		
CDC19 PAS_chr2-1_0769	Adr1(1) [°] , Cat8(2), Sip4(6), Hap2/3/4/5(1), Rds2(2), Ert1(5), Stb5(2), Msn2(3) ^{°*} , Msn4(3) ^{°*} , Mig1(3) [°] , Gcr1(3) ^{°*}	Cat8(2), Sip4(2), Stb5(1), Msn2(5), Msn4(2), Mig1(1)	Sip4(2), Ert1(1), Stb5(1)
PCK1 PAS_FragB_0061	Cat8(4) ^{°*} , Sip4(3) ^{°*} , Hap2/3/4/5(1) [°] , Rds2(1) ^{°*} , Ert1(3)*, Msn2(1) ^{°*} , Msn4(1) ^{°*} , Mig1(3), Gcr1(3) [°]	Adr1(1), Cat8(4), Ert1(3), Stb5(2), Msn2(2), Msn4(2), Mig1(5)	—

Color Codings: Light blue for Glycolysis exclusive genes, Deep blue for Gluconeogenesis exclusive genes, Violet if common genes for both pathways

Table C.2: Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on *S. cerevisiae* and *P. pastoris* promoters that regulate expression of enzymes that act on Pentose Phosphate Pathway.

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
ZWF1 PAS_chr2-1_0308	Adr1(1)*, Cat8(1), Sip4(3), Hap2/3/4/5(1), Rds2(1)*, Ert1(5), Stb5(3)*, Msn2(5) ^{°*} , Msn4(5) ^{°*} , Mig1(8), GCR1(1) [°]	Adr1(1), Cat8(4), Sip4(2), Ert1(4), Stb5(1), Msn2(5), Msn4(4), Mig1(2)	Msn2(1), Msn4(1)
SOL3 PAS_chr3_1126	Cat8(1), Rds2(1), Stb5(2) ^{°*} , Msn2(1)*, Msn4(1), GCR1(1) [°]	Adr1(3)	Msn4(1)
SOL4 PAS_chr3_1126	Sip4(2), Hap2/3/4/5(1) [°] , Rds2(1), Ert1(2), Msn2(1) ^{°*} , Msn4(1) [°] , Mig1(3) [°]		
Continued on next page			

Table C.2 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
GND1 PAS_chr3_0277	Adr1(1), Cat8(1), Hap2/3/4/5(1), Rds2(2), Stb5(3) ^{°*} , Msn2(1) ^{°*} , Msn4(1)*, Mig1(1)	Adr1(1), Cat8(2), Stb5(1), Msn2(1), Msn4(2)	—
RKI1 PAS_chr4_0213	Cat8(2) [°] , Sip4(2)*, Ert1(3), Stb5(3)*, Msn2(2) [°] , Msn4(2) ^{°*} , Mig1(3) [°]	Adr1(2), Cat8(1), Stb5(1), Msn2(3), Msn4(1), Mig1(1)	Msn2(1)
RPE1 AT250_GQ6803479	Sip4(2), Hap2/3/4/5(2), Ert1(2), Mig1(2)	Adr1(1), Ert1(3), Stb5(1), Msn2(2), Mig1(1)	—
TKL1 PAS_chr1-4_0150	Sip4(2), Rds2(1)*, Ert1(3)*, Stb5(3) ^{°*} , Msn2(1)*, Msn4(1)*, Mig1(1)	Cat8(2), Sip4(2), Msn2(3), Msn4(1), Mig1(1)	Msn4(1)
Continued on next page			

Table C.2 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
TAL1 PAS_chr2-2_0337	Stb5(2) ^o *, Msn2(1)*, Msn4(1)*, Mig1(1), GCR1(6)	Cat8(1), Ert1(1), Stb5(1), Msn2(2), Msn4(1)	Msn4(1)

Color Codings: Light green for Pentose Phosphate Pathway

Table C.3: Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on *S. cerevisiae* and *P. pastoris* promoters that regulate expression of enzymes that act on TCA and Glyoxylate Cycles.

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
PDA1 PAS_chr2-2_0294	Adr1(1), Hap2/3/4/5(1), Stb5(2) ^o , Msn2(2), Msn4(2)	Cat8(5), Ert1(1), Stb5(1), Msn2(5), Msn4(4), Mig1(3)	Msn2(2), Msn4(1)
PDB1 PAS_chr1-4_0593	Sip4(2), Ert1(2), Stb5(2) ^o , Msn2(2), Msn4(2)	Cat8(1), Ert1(1), Stb5(1), Msn2(3), Msn4(1), Mig1(1)	—

Continued on next page

Table C.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
LAT1 PAS_chr1-1_0050	Cat8(1), Hap2/3/4/5(1)*, Rds2(1), Ert1(1), Msn4(1), Mig1(2)	Adr1(3), Cat8(7), Ert1(2), Stb5(1), Msn2(2), Msn4(1), Mig1(4)	Cat8(1), Msn4(1)
LPD1 PAS_chr2-2_0048	Sip4(1), Hap2/3/4/5(2)°, Msn2(2), Msn4(2)*,	Adr1(1), Cat8(4), Ert1(2), Msn2(4)	—
PDX1 PAS_chr1-4_0254	Sip4(1), Rds2(1), Stb5(2), Msn2(2)°, Msn4(2)°	Ert1(1), Msn2(1)	—
ACO1 PAS_chr1-3_0104	Adr1(1), Cat8(2), Sip4(3), Hap2/3/4/5(2)°, Rds2(1), Ert1(1), Stb5(2), Msn2(1)*, Msn4(1)*, Mig1(2), GCR1(3)°	Adr1(1), Cat8(4), Hap2/3/4/5(1), Stb5(2), Msn2(1), Msn4(3), Mig1(1)	Adr1(1), Hap2/3/4/5(1)
Continued on next page			

Table C.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
CIT1 PAS_chr1-1_0475	Cat8(1), Sip4(2), Hap2/3/4/5(1) ^{°*} , Rds2(1)*, Ert1(5), Msn2(3) ^{°*} , Msn4(3) ^{°*} , Mig1(2), GCR1(1)*	Cat8(4), Ert1(3), Stb5(2), Msn2(2), Msn4(2), Mig1(3)	Ert1(2), Msn2(1), Msn4(1)
CIT2 PAS_chr1-1_0475	Adr1(1), Cat8(1) ^{°*} , Sip4(1), Hap2/3/4/5(2), Stb5(1), Msn2(2) ^{°*} , Msn4(2) ^{°*} , GCR1(2) [°]		
ICL1 PAS_chr1-4_0338	Cat8(1) ^{°*} , Hap2/3/4/5(1) ^{°*} , Rds2(1), Ert1(1), GCR1(2)	Cat8(4), Stb5(3), Msn2(3), Msn4(1), Mig1(1)	—
MLS1 PAS_chr4_0191	Adr1(1) ^{°*} , Cat8(6) ^{°*} , Sip4(5) ^{°*} , Hap2/3/4/5(1) [°] , Ert1(4), Stb5(6), Msn2(3) ^{°*} , Msn4(3) ^{°*} , GCR1(1)	Adr1(1), Cat8(5), Ert1(5), Stb5(2), Msn2(4), Msn4(2), Mig1(1)	Cat8(1), Msn2(1), Msn4(1)
Continued on next page			

Table C.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
IDH1 PAS_chr4_0580	Cat8(1) [°] , Hap2/3/4/5(4) [°] , Stb5(2) [°] , Msn2(2) ^{°*} , Msn4(2) ^{°*} , Mig1(2)	Cat8(1), Sip4(2), Stb5(1), Msn2(3), Msn4(1)	Msn2(1), Msn4(1)
IDH2 PAS_chr2-1_0120	Adr1(1), Cat8(1), Sip4(2), Hap2/3/4/5(1), Stb5(2), Msn2(2) ^{°*} , Msn4(2)*, Mig1(1)	Adr1(1), Cat8(5), Stb5(2), Msn2(3), Msn4(2), Mig1(1)	Stb5(1)
KGD1 PAS_chr2-1_0089	Cat8(1), Sip4(2), Ert1(2), Stb5(3), Msn2(2)*, Msn4(2)	Cat8(6), Stb5(1), Msn2(3), Msn4(1), Mig1(3)	—
KGD2 PAS_chr1-3_0094	Cat8(2), Sip4(4), Hap2/3/4/5(2) ^{°*} , Ert1(4), Msn2(1), Msn4(1), GCR1(1)	Adr1(2), Cat8(4), Ert1(2), Stb5(1), Mig1(3)	—
Continued on next page			

Table C.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
LSC1 PAS_chr3_0831	Sip4(1), Hap2/3/4/5(1) [°] , Ert1(2), Msn2(2), Msn4(2), Mig1(1), GCR1(2)	Cat8(5), Stb5(3), Msn2(4), Msn4(4), Mig1(1)	—
LSC2 PAS_chr2-2_0407	Adr1(1)*, Sip4(3), Rds2(3)*, Ert1(1)*, Stb5(1), Msn2(1) [°] , Msn4(1) [°] , Mig1(1)	Adr1(1), Ert1(2), Stb5(1), Msn2(4), Msn4(3), Mig1(1)	—
SDH1 PAS_chr4_0733	Cat8(2) [°] , Hap2/3/4/5(2) ^{°*} , Rds2(1), Ert1(2), Stb5(1), Msn2(1)*, Msn4(1)*, Mig1(2), GCR1(1) [°]	Adr1(1), Cat8(4), Hap2/3/4/5(1), Ert1(2), Stb5(1), Msn2(1), Mig1(1)	Hap2/3/4/5(1)
SDH2 PAS_chr3_1111	Hap2/3/4/5(2) [°] , Stb5(2), Msn2(1) [°] , Msn4(1) [°]	Adr1(1), Cat8(1), Stb5(1)	Stb5(1)
Continued on next page			

Table C.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
SDH3 PAS_chr1-4_0487	Hap2/3/4/5(2) [°] , Mig1(1), GCR1(1) [°]	Ert1(1), Msn2(3), Msn4(2)	—
SDH4 PAS_chr2-2_0283	Sip4(1), Hap2/3/4/5(1) [°] , Ert1(2), Msn2(1), Msn4(1), Mig1(1)	Adr1(2), Stb5(1), Msn2(1), Msn4(1), Mig1(2)	—
FUM1 PAS_chr3_0647	Adr1(1), Cat8(2) [°] , Sip4(4), Hap2/3/4/5(2), Rds2(1), Ert1(3), Msn2(2) [°] , Msn4(2) [°] , Mig1(1), GCR1(1) [°]	Adr1(3), Ert1(2), Stb5(1), Msn2(1), Msn4(1), Mig1(3)	Adr1(1), Msn2(1), Msn4(1)
MDH1 PAS_chr2-1_0238	Cat8(1), Sip4(4), Hap2/3/4/5(1) [°] , Ert1(3)*, Stb5(2), Msn2(2) [°] *, Msn4(2) [°] *, Mig1(7), GCR1(1) [°]	Adr1(2), Cat8(1), Sip4(2), Ert1(1), Stb5(1), Msn2(2), Msn4(1)	Msn2(1)
Continued on next page			

Table C.3 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
MDH2 PAS_chr4_0815	Adr1(1) ^{°*} , Cat8(4) ^{°*} , Sip4(2)*, Hap2/3/4/5(1) [°] , Rds2(1) [°] , Ert1(6), Stb5(2) [°] , Msn2(4) ^{°*} , Msn4(4) ^{°*} , GCR1(1) [°]	Cat8(1), Ert1(1), Msn2(5), Msn4(3), Mig1(1)	Msn2(1), Msn4(2)

Color Codings: Red for TCA cycle exclusive genes, Pink for Glyoxylate cycle exclusive genes, Purple for common genes for both pathways.

Table C.4: Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on *S. cerevisiae* and *P. pastoris* promoters that regulate expression of enzymes that act on Ethanol Utilization and Alcoholic Fermentation Pathways.

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
PDC1 PAS_chr3_0188	Cat8(1), Sip4(4), Ert1(2)*, Msn2(1) ^{°*} , Msn4(1) ^{°*} , Mig1(2)*, GCR1(2) ^{°*}	Ert1(1), Stb5(2), Msn2(1), Msn4(1), Mig1(1),	—

Continued on next page

Table C.4 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
ADH1 PAS_chr2-1_0472	Sip4(1), Hap2/3/4/5(1) ^{°*} , Rds2(1) [°] , Msn2(3)*, Msn4(4)*, Mig1(5) [°] , GCR1(4) ^{°*}	Adr1(2), Cat8(5), Ert1(4), Stb5(2), Msn2(18), Msn4(13), Mig1(8)	Msn2(5), Msn4(5), Mig1(1),
ADH2 PAS_chr2-1_0472	Adr1(2) ^{°*} , Cat8(3) ^{°*} , Sip4(8) [°] , Rds2(1) [°] , Ert1(7), Msn2(2)*, Msn4(2)*,		
ALD4 PAS_chr4_0043	Adr1(2) ^{°*} , Cat8(2) [°] , Hap2/3/4/5(3) [°] , Ert1(1), Stb5(1)*, Msn2(3) ^{°*} , Msn4(3) ^{°*} , Mig1(6), GCR1(3) [°]	Adr1(2), Cat8(3), Stb5(1), Msn2(2),	Adr1(1), Msn4(2)
ALD6 PAS_chr3_0987	Cat8(3) [°] , Stb5(4) ^{°*} , Msn2(4) ^{°*} , Msn4(4) ^{°*} , Mig1(6) [°] , GCR1(1) [°]	Adr1(1), Cat8(18), Sip4(2), Ert1(2), Stb5(1), Msn2(5), Msn4(5), Mig1(2)	—
Continued on next page			

Table C.4 – continued from previous page

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
ACS1 PAS_chr2-1_0767	Cat8(3) [°] *, Sip4(2) [°] , Rds2(1), Stb5(2), Msn2(3)*, Msn4(3)*, Mig1(5)	Cat8(3), Ert1(1), Msn2(6), Msn4(4)	Cat8(1), Msn2(2),
ACS2 PAS_chr3_0403	Sip4(2), Hap2/3/4/5(2), Ert1(1), Stb5(1), Msn2(1) [°] , Msn4(1), Mig1(1), GCR1(3) [°]	Cat8(3), Stb5(1), Msn2(4), Mig1(2)	—

Color Codings: Gray for Alcoholic Fermentation Pathway genes, White for Ethanol Utilization Pathway genes.

Table C.5: Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on *S. cerevisiae* and *P. pastoris* promoters that regulate expression of enzymes that act on Glycerol Utilization Pathway.

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
GUT1 PAS_chr4_0783	Adr1(1) ^{°*} , Sip4(1), Hap2/3/4/5(2), Ert1(2), Stb5(1), Msn2(3), Msn4(3), Mig1(3) [°] , GCR1(2)	Cat8(7), Ert1(3), Msn2(6), Msn4(6), Mig1(4)	Msn2(2), Msn4(1)
GUT2 PAS_chr3_0579	Cat8(1), Sip4(1), Rds2(1), Stb5(1) [°] , Msn2(1) [°] , Msn4(1) [°] , Mig1(1) [°]	Cat8(4), Sip4(2), Ert1(2), Msn2(10), Msn4(9), Mig1(6)	Msn4(1)

Color Codings: Yellow for Glycerol Utilization Pathways

Table C.6: Number of TFBSs annotated for Adr1, Cat8, Sip4, Hap2/3/4/5, Rds2, Ert1, Stb5, Msn2, Msn4, Mig1, Tye7 and Gcr1 TFs on *S. cerevisiae* and *P. pastoris* promoters that regulate expression of enzymes that act on Anaplerotic Reactions.

Gene ID	<i>S. cerevisiae</i> (Scan)	<i>P. pastoris</i> (Scan)	<i>P. pastoris</i> (Footprinting)
MAE1 PAS_chr3_0181	Cat8(2), Sip4(2), Ert1(1) ^{°*} , Msn2(5) ^{°*} , Msn4(4) ^{°*}	Cat8(2), Sip4(2), Ert1(1), Msn2(5), Msn4(4)	—
PYC1 PAS_chr2-2_0024	Cat8(2), Sip4(5) [°] , Rds2(1)*, Ert1(2)*, Stb5(2) [°] , Msn2(1) ^{°*} , Msn4(1) ^{°*} , Mig1(3), GCR1(1) [°]	Cat8(3), Ert1(5), Stb5(1), Msn2(4), Msn4(4), Mig1(3),	Cat8(1), Msn2(1), Msn4(1),
PYC2 PAS_chr2-2_0024	Sip4(1), Hap2/3/4/5(1), Rds2(2), Stb5(2), Msn2(3)*, Msn4(3)*, Mig1(1), GCR1(2)		

Color Codings: Bronze for Anaplerotic Reactions



APPENDIX D

EFFECT OF REMOVING FEATURES ON MODEL PERFORMANCES



Table D.1: Changes of MCC Scores of the models as number of features is decreased

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
BAD08_Tea1	—	0.751	4	0.749	—	—	—	—	—	—	—	—
BAD08_Ace2	—	0.829	0	0.858	2	0.881	4	0.880	—	—	—	—
BAD08_Ynr063w	—	0.764	4	0.764	—	—	—	—	—	—	—	—
BAD08_Yap3	—	0.686	2	0.721	0	0.743	1	0.821	4	0.886	—	—
BAD08_Rgt1	—	0.779	0	0.773	—	—	—	—	—	—	—	—
BAD08_Xbp1	—	0.969	1	0.974	4	0.974	—	—	—	—	—	—
BAD08_Rgm1	—	0.858	0	0.903	4	0.897	—	—	—	—	—	—
BAD08_Rph1	—	0.889	0	0.904	1	0.947	2	0.942	—	—	—	—
BAD08_Phdl	—	0.757	2	0.783	4	0.783	—	—	—	—	—	—
BAD08_Sum1	—	0.857	0	0.874	4	0.904	1	0.910	2	0.932	—	—
BAD08_Rfx1	—	0.661	0	0.720	4	0.734	1	0.794	2	0.886	—	—
BAD08_Stb4	—	0.795	2	0.813	4	0.813	—	—	—	—	—	—
BAD08_Phoh	—	0.855	0	0.926	4	0.937	2	0.942	1	0.931	—	—
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed									
	0		1		2		3		4	
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
BAD08_Uga3	—	0.804	2	0.827	4	0.827	—	—	—	—
BAD08_Ygr067c	—	0.864	1	0.886	0	0.892	4	0.903	2	0.910
BAD08_Mig1	—	0.882	4	0.882	—	—	—	—	—	—
BAD08_Sip4	—	0.721	4	0.721	—	—	—	—	—	—
BAD08_Hac1	—	0.673	2	0.713	0	0.716	4	0.789	1	0.914
BAD08_Mbp1	—	0.936	2	0.936	4	0.936	—	—	—	—
BAD08_Dal80	—	0.621	0	0.669	4	0.761	1	0.903	2	0.925
BAD08_Rds2	—	0.823	4	0.823	—	—	—	—	—	—
BAD08_Skn7	—	0.852	4	0.852	—	—	—	—	—	—
BAD08_Yjl103c	—	0.768	0	0.789	4	0.887	1	0.908	2	0.925
BAD08_Yrm1	—	0.708	1	0.745	2	0.725	—	—	—	—
BAD08_Rei1	—	0.840	0	0.864	4	0.926	2	0.920	—	—
BAD08_Gat4	—	0.748	2	0.749	0	0.767	4	0.859	1	0.893
BAD08_Yer130c	—	0.920	1	0.921	0	0.931	2	0.936	4	0.969
Continued on next page										

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
BAD08_Cbf1	—	0.552	2	0.621	0	0.668	1	0.800	4	0.862		
BAD08_Gat3	—	0.845	0	0.857	4	0.908	1	0.942	2	0.942		
BAD08_Rsc3	—	0.795	2	0.827	0	0.827	4	0.863	1	0.909		
BAD08_Hmra2	—	0.885	0	0.897	4	0.908	2	0.914	1	0.914		
BAD08_Put3	—	0.815	0	0.815	—	—	—	—	—	—		
BAD08_Stb5	—	0.703	0	0.728	4	0.795	1	0.868	2	0.897		
BAD08_Dot6	—	0.818	0	0.852	4	0.903	1	0.931	2	0.925		
BAD08_Ume6	—	0.785	2	0.804	1	0.814	4	0.814	—	—		
BAD08_Lys14	—	0.778	4	0.778	—	—	—	—	—	—		
BAD08_Ecm23	—	0.848	2	0.853	4	0.853	—	—	—	—		
BAD08_Tos8	—	0.783	0	0.801	2	0.868	4	0.862	—	—		
BAD08_Tbf1	—	0.818	0	0.851	4	0.868	1	0.920	2	0.931		
BAD08_Ylr278c	—	0.834	2	0.851	4	0.851	—	—	—	—		
BAD08_Abf1	—	0.286	0	0.500	1	0.527	2	0.789	4	0.903		
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0			1			2			3		
	REM	MCC		REM	MCC		REM	MCC		REM	MCC	
BAD08_Cst6	—	0.644		2	0.674		0	0.736		1	0.809	
BAD08_Nhp10	—	0.554		0	0.602		4	0.618		1	0.702	
BAD08_Sig1	—	0.881		0	0.904		2	0.910		1	0.937	
BAD08_Cin5	—	0.707		0	0.794		4	0.844		1	0.891	
BAD08_Fkh2	—	0.614		1	0.664		0	0.713		4	0.801	
BAD08_Ydr520c	—	0.777		1	0.813		4	0.813		—	—	
BAD08_Rpn4	—	0.657		0	0.684		4	0.706		1	0.857	
BAD08_Ph02	—	0.810		2	0.827		0	0.852		4	0.881	
BAD08_Azf1	—	0.698		2	0.727		4	0.727		—	—	
BAD08_Sok2	—	0.587		0	0.629		1	0.720		2	0.813	
BAD08_Tbs1	—	0.780		0	0.807		4	0.837		1	0.891	
BAD08_Stp3	—	0.821		4	0.821		—	—		—	—	
BAD08_Dal82	—	0.818		0	0.821		4	0.852		1	0.880	
BAD08_Abf2	—	0.689		2	0.719		4	0.719		—	—	
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
BAD08_Stp4	—	0.809	0	0.826	4	0.903	1	0.942	2	0.943	—	—
BAD08_Mig3	—	0.820	1	0.860	4	0.860	—	—	—	—	—	—
BAD08_Rim101	—	0.618	0	0.669	4	0.677	1	0.874	2	0.914	—	—
BAD08_Cha4	—	0.577	0	0.626	4	0.670	1	0.886	2	0.914	—	—
BAD08_Yrr1	—	0.840	4	0.840	—	—	—	—	—	—	—	—
BAD08_Hcm1	—	0.753	4	0.753	—	—	—	—	—	—	—	—
BAD08_Rsc30	—	0.793	0	0.813	2	0.841	4	0.863	1	0.947	—	—
BAD08_Pdr8	—	0.764	0	0.785	4	0.869	1	0.888	2	0.880	—	—
BAD08_Cep3	—	0.855	0	0.859	4	0.926	2	0.942	1	0.942	—	—
BAD08_Aft2	—	0.791	3	0.801	2	0.819	4	0.819	—	—	—	—
BAD08_Hsf1	—	0.726	4	0.745	0	0.802	2	0.814	1	0.898	—	—
BAD08_Pdr1	—	0.773	2	0.809	4	0.809	—	—	—	—	—	—
BAD08_Msn4	—	0.835	1	0.870	4	0.870	—	—	—	—	—	—
BAD08_Ypr022c	—	0.838	2	0.844	0	0.850	4	0.840	—	—	—	—
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed														
	0			1			2			3			4		
	REM	MCC		REM	MCC		REM	MCC		REM	MCC		REM	MCC	
BAD08_Ypr013c	—	0.852		1	0.862		2	0.868		4	0.868		—	—	
BAD08_Swi5	—	0.540		0	0.604		4	0.685		1	0.845		2	0.880	
BAD08_Gzf3	—	0.781		0	0.840		4	0.851		1	0.931		2	0.936	
BAD08_Srd1	—	0.845		0	0.880		4	0.909		1	0.947		2	0.958	
BAD08_Rox1	—	0.871		1	0.893		2	0.900		4	0.900		—	—	
BAD08_Yer184c	—	0.595		2	0.614		0	0.616		4	0.749		1	0.862	
BAD08_Crz1	—	0.797		4	0.797		—	—		—	—		—	—	
BAD08_Cat8	—	0.880		2	0.891		0	0.898		1	0.892		—	—	
BAD08_Asg1	—	0.828		1	0.841		4	0.841		—	—		—	—	
BAD08_Rdr1	—	0.776		2	0.798		1	0.804		4	0.782		—	—	
BAD08_Yml081w	—	0.767		3	0.800		0	0.790		—	—		—	—	
BAD08_Tec1	—	0.715		0	0.768		4	0.814		1	0.863		2	0.892	
BAD08_Ypr196w	—	0.737		1	0.820		4	0.820		—	—		—	—	
BAD08_Hap1	—	0.610		2	0.622		4	0.621		—	—		—	—	
Continued on next page															

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
BAD08_Msn2	—	0.840	0	0.857	4	0.886	1	0.887	2	0.943		
BAD08_Eds1	—	0.701	1	0.713	0	0.734	4	0.862	2	0.886		
BAD08_Reb1	—	0.493	2	0.600	1	0.656	0	0.807	4	0.869		
BAD08_Ykl222c	—	0.672	4	0.732	0	0.796	1	0.908	2	0.903		
BAD08_Fzf1	—	0.714	0	0.775	1	0.832	4	0.868	2	0.931		
BAD08_Gis1	—	0.893	2	0.915	4	0.915	—	—	—	—		
BAD08_Gln3	—	0.887	0	0.903	4	0.915	1	0.932	2	0.948		
BAD08_Yox1	—	0.932	2	0.942	4	0.942	—	—	—	—		
BAD08_Mig2	—	0.770	2	0.771	0	0.777	4	0.804	1	0.893		
BAD08_Ybr239c	—	0.852	2	0.875	4	0.875	—	—	—	—		
BAD08_Rds1	—	0.778	0	0.803	4	0.876	1	0.909	2	0.958		
BAD08_Met31	—	0.785	0	0.821	1	0.828	4	0.898	2	0.898		
BAD08_Tye7	—	0.579	2	0.613	0	0.684	1	0.775	4	0.891		
BAD08_Zms1	—	0.534	0	0.586	1	0.680	4	0.834	2	0.875		
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed									
	0		1		2		3		4	
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
BAD08_Cup9	—	0.817	0	0.847	2	0.865	4	0.855	—	—
BAD08_Yb1054w	—	0.525	0	0.707	2	0.794	1	0.850	4	0.891
BAD08_Ypl230w	—	0.871	0	0.904	4	0.937	2	0.931	—	—
BAD08_Adr1	—	0.829	2	0.843	4	0.843	—	—	—	—
BAD08_Ste12	—	0.523	0	0.645	2	0.706	1	0.807	4	0.891
BAD08_Oaf1	—	0.661	0	0.676	4	0.880	1	0.881	2	0.908
BAD08_Fhl1	—	0.803	0	0.815	4	0.840	1	0.898	2	0.914
BAD08_Swi4	—	0.896	0	0.943	2	0.958	4	0.953	—	—
BAD08_Hal9	—	0.838	4	0.838	—	—	—	—	—	—
BAD08_Gat1	—	0.789	0	0.864	4	0.875	1	0.953	2	0.953
BAD08_Y11054c	—	0.518	2	0.518	0	0.552	4	0.713	1	0.868
NBT06_Ceh-22	—	0.844	4	0.844	—	—	—	—	—	—
NBT06_Oct-1	—	0.869	2	0.869	—	—	—	—	—	—
NBT06_Cbf1	—	0.864	0	0.886	4	0.931	2	0.926	—	—
Continued on next page										

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
NBT06_Zif268	—	0.854	1	0.860	0	0.866	4	0.931	2	0.931	2	0.931
NBT06_Rap1	—	0.626	2	0.638	0	0.644	1	0.786	4	0.886	4	0.886
NAR10_Gcn4	—	0.858	0	0.884	4	0.932	2	0.933	1	0.918	1	0.918
NAR10_Junfos	—	0.410	0	0.535	4	0.597	1	0.734	2	0.868	2	0.868
ZHU09_Rds2-9	—	0.820	0	0.849	4	0.926	2	0.915	—	—	—	—
ZHU09_Leu3	—	0.948	0	0.958	2	0.969	1	0.974	4	0.953	4	0.953
ZHU09_Sfp1-9	—	0.865	2	0.899	4	0.899	—	—	—	—	—	—
ZHU09_Sip4-9	—	0.749	2	0.761	4	0.761	—	—	—	—	—	—
ZHU09_Gzf3-9	—	0.822	0	0.883	4	0.902	2	0.933	1	0.936	1	0.936
ZHU09_Ume6-9	—	0.909	4	0.909	—	—	—	—	—	—	—	—
ZHU09_Sum1-9	—	0.840	0	0.874	4	0.886	1	0.910	2	0.958	2	0.958
ZHU09_Tec1	—	0.860	2	0.878	4	0.878	—	—	—	—	—	—
ZHU09_Ph04-9	—	0.877	0	0.942	2	0.948	4	0.948	—	—	—	—
ZHU09_Bas1	—	0.613	0	0.727	1	0.794	4	0.856	2	0.908	2	0.908
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed									
	0		1		2		3		4	
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
ZHU09_Ygr067c	—	0.744	1	0.752	0	0.752	—	—	—	—
ZHU09_Yer130c-11	—	0.721	2	0.755	0	0.769	4	0.814	1	0.870
ZHU09_Rds1	—	0.798	2	0.810	4	0.810	—	—	—	—
ZHU09_Gcn4-9	—	0.871	0	0.897	4	0.958	2	0.958	—	—
ZHU09_Mig2	—	0.898	4	0.898	—	—	—	—	—	—
ZHU09_Yox1	—	0.927	2	0.943	4	0.943	—	—	—	—
ZHU09_Matalpha2-9	—	0.916	1	0.917	4	0.917	—	—	—	—
ZHU09_Gln3	—	0.835	0	0.886	4	0.925	1	0.963	2	0.963
ZHU09_Gat1-11	—	0.870	0	0.910	4	0.969	1	0.958	—	—
ZHU09_Ypr013c-9	—	0.832	4	0.832	—	—	—	—	—	—
ZHU09_Sip4-11	—	0.722	4	0.722	—	—	—	—	—	—
ZHU09_Ndt80-9	—	0.570	4	0.570	—	—	—	—	—	—
ZHU09_Put3-11	—	0.806	1	0.832	4	0.832	—	—	—	—
ZHU09_Nhp6b-9	—	0.898	4	0.898	—	—	—	—	—	—
Continued on next page										

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
ZHU09_Cup9	—	0.663	0	0.668	4	0.762	2	0.809	1	0.897	—	—
ZHU09_Srd1-11	—	0.771	2	0.790	0	0.820	4	0.832	1	0.931	—	—
ZHU09_Matalpha2-11	—	0.847	4	0.847	—	—	—	—	—	—	—	—
ZHU09_Hal9	—	0.814	0	0.824	4	0.899	2	0.915	1	0.936	—	—
ZHU09_Cbf1-11	—	0.870	0	0.942	4	0.969	2	0.984	1	0.931	—	—
ZHU09_Ymr063w	—	0.752	4	0.752	—	—	—	—	—	—	—	—
ZHU09_Pbf1-9	—	0.894	0	0.900	4	0.942	1	0.947	2	0.947	—	—
ZHU09_Gsm1-9	—	0.816	2	0.834	4	0.834	—	—	—	—	—	—
ZHU09_Cha4	—	0.773	4	0.773	—	—	—	—	—	—	—	—
ZHU09_Sfl1	—	0.655	2	0.668	0	0.670	4	0.781	1	0.903	—	—
ZHU09_Ypr015c	—	0.665	0	0.689	1	0.757	4	0.847	2	0.856	—	—
ZHU09_Ypr013c-11	—	0.772	1	0.811	2	0.840	4	0.840	—	—	—	—
ZHU09_Mig3	—	0.827	1	0.845	4	0.845	—	—	—	—	—	—
ZHU09_Stp4	—	0.716	0	0.761	2	0.810	4	0.805	—	—	—	—
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed									
	0		1		2		3		4	
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
ZHU09_Met32-9	—	0.793	0	0.816	4	0.845	1	0.892	2	0.920
ZHU09_Gzf3-11	—	0.875	0	0.903	4	0.920	1	0.953	2	0.953
ZHU09_Rds2-11	—	0.861	4	0.861	—	—	—	—	—	—
ZHU09_Nhp6a-9	—	0.806	2	0.835	0	0.842	1	0.859	4	0.890
ZHU09_Pbf2-11	—	0.821	0	0.848	4	0.915	2	0.937	1	0.926
ZHU09_Ydr520c	—	0.593	1	0.669	0	0.748	4	0.862	2	0.886
ZHU09_Pdr1	—	0.529	2	0.555	4	0.555	—	—	—	—
ZHU09_Pbf2-9	—	0.834	0	0.843	4	0.908	2	0.925	1	0.904
ZHU09_Fkh2-11	—	0.723	0	0.757	1	0.755	—	—	—	—
ZHU09_Cep3	—	0.743	0	0.774	4	0.887	2	0.887	—	—
ZHU09_Put3-9	—	0.705	0	0.724	4	0.851	1	0.851	—	—
ZHU09_Fhl1-11	—	0.800	0	0.900	4	0.905	1	0.927	2	0.937
ZHU09_Mgal	—	0.752	2	0.751	—	—	—	—	—	—
ZHU09_Nrg1-11	—	0.832	0	0.870	4	0.887	1	0.926	2	0.942
Continued on next page										

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0			1			2			3		
	REM	MCC		REM	MCC		REM	MCC		REM	MCC	
ZHU09_Rpn4-9	—	0.545	0	0.659	1	0.743	2	0.839	4	0.880		
ZHU09_Nhp6b-11	—	0.859	2	0.885	0	0.902	1	0.918	4	0.934		
ZHU09_Fhl1-9	—	0.776	0	0.781	1	0.851	4	0.898	2	0.915		
ZHU09_Rsc30	—	0.886	2	0.898	4	0.898	—	—	—	—		
ZHU09_Gat1-9	—	0.844	0	0.898	4	0.942	2	0.963	1	0.942		
ZHU09_Rtg3	—	0.596	0	0.604	1	0.661	2	0.850	4	0.897		
ZHU09_Asg1	—	0.780	1	0.805	2	0.821	4	0.821	—	—		
ZHU09_Yap6	—	0.604	0	0.741	4	0.782	1	0.815	2	0.898		
ZHU09_Sum1-11	—	0.899	0	0.916	2	0.918	4	0.933	1	0.958		
ZHU09_Tye7-9	—	0.814	0	0.880	4	0.920	2	0.903	—	—		
ZHU09_Rgt1-9	—	0.769	3	0.777	4	0.777	—	—	—	—		
ZHU09_Yap1	—	0.798	0	0.862	4	0.880	2	0.880	1	0.892		
ZHU09_Ume6-11	—	0.864	4	0.864	—	—	—	—	—	—		
ZHU09_Rsc3	—	0.922	1	0.932	2	0.933	4	0.933	—	—		
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4		REM	MCC
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC		
ZHU09_Gat3	—	0.931	4	0.931	—	—	—	—	—	—	—	—
ZHU09_Sut2-9	—	0.790	0	0.818	4	0.863	1	0.886	2	0.909	—	—
ZHU09_Fkh1-9	—	0.646	1	0.654	2	0.659	4	0.659	—	—	—	—
ZHU09_Yrm1	—	0.576	2	0.610	0	0.618	1	0.721	4	0.863	—	—
ZHU09_Yml081w	—	0.834	4	0.834	—	—	—	—	—	—	—	—
ZHU09_Aro80	—	0.766	2	0.817	4	0.795	—	—	—	—	—	—
ZHU09_Ndt80-11	—	0.534	1	0.576	0	0.606	4	0.816	2	0.859	—	—
ZHU09_Rap1-11	—	0.605	4	0.605	—	—	—	—	—	—	—	—
ZHU09_Tbf1	—	0.817	0	0.840	4	0.868	1	0.937	2	0.937	—	—
ZHU09_Cbf1-9	—	0.844	0	0.904	4	0.925	2	0.914	—	—	—	—
ZHU09_Nhp6a-11	—	0.837	2	0.859	0	0.882	4	0.878	—	—	—	—
ZHU09_Rph1-11	—	0.703	0	0.769	1	0.829	2	0.881	4	0.903	—	—
ZHU09_Yll054c-11	—	0.668	0	0.685	4	0.801	1	0.838	2	0.868	—	—
ZHU09_Stp2-9	—	0.782	0	0.859	4	0.883	2	0.894	1	0.909	—	—
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
ZHU09_Yrr1-9	—	0.670	2	0.701	4	0.701	—	—	—	—	—	—
ZHU09_Usv1	—	0.834	0	0.863	4	0.903	2	0.920	1	0.903	—	—
ZHU09_Rph1-9	—	0.858	0	0.881	1	0.915	4	0.931	2	0.942	—	—
ZHU09_Yrr1-11	—	0.738	0	0.747	4	0.838	1	0.886	2	0.875	—	—
ZHU09_Fkh1-11	—	0.672	2	0.692	4	0.692	—	—	—	—	—	—
ZHU09_Oaf1-11	—	0.437	2	0.454	0	0.498	4	0.662	1	0.885	—	—
ZHU09_Lys14	—	0.752	3	0.751	—	—	—	—	—	—	—	—
ZHU09_Gat4-11	—	0.791	2	0.816	4	0.816	—	—	—	—	—	—
ZHU09_Mcm1-11	—	0.713	0	0.741	1	0.814	4	0.852	2	0.886	—	—
ZHU09_Sfp1-11	—	0.822	0	0.868	1	0.882	4	0.898	2	0.915	—	—
ZHU09_Met32-11	—	0.828	4	0.828	—	—	—	—	—	—	—	—
ZHU09_Pbf1-11	—	0.875	2	0.893	4	0.893	—	—	—	—	—	—
ZHU09_Ph02	—	0.886	4	0.886	—	—	—	—	—	—	—	—
ZHU09_Aft1	—	0.527	1	0.566	4	0.566	—	—	—	—	—	—
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed									
	0		1		2		3		4	
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
ZHU09_Stp2-11	—	0.887	0	0.915	4	0.920	2	0.925	1	0.914
ZHU09_Y11054c-9	—	0.683	4	0.683	—	—	—	—	—	—
ZHU09_Gal4-11	—	0.800	2	0.813	4	0.813	—	—	—	—
ZHU09_Gsm1-11	—	0.880	4	0.880	—	—	—	—	—	—
ZHU09_Rdr1-9	—	0.807	2	0.865	1	0.818	—	—	—	—
ZHU09_Xbp1	—	0.918	0	0.927	4	0.927	—	—	—	—
ZHU09_Mcm1-9	—	0.630	1	0.716	2	0.771	0	0.816	4	0.909
ZHU09_Rpn4-11	—	0.534	0	0.647	2	0.700	1	0.814	4	0.862
ZHU09_Gat4-9	—	0.795	2	0.819	0	0.828	4	0.893	1	0.915
ZHU09_Smp1	—	0.678	1	0.707	0	0.723	4	0.886	2	0.898
ZHU09_Ph04-11	—	0.868	0	0.914	4	0.947	2	0.958	1	0.942
ZHU09_Ykl222c-11	—	0.706	2	0.736	4	0.736	—	—	—	—
ZHU09_Tea1	—	0.721	1	0.741	4	0.741	—	—	—	—
ZHU09_Oaf1-9	—	0.762	2	0.802	4	0.802	—	—	—	—
Continued on next page										

Table D.1 – continued from previous page

	Number of Features Removed														
	0			1			2			3			4		
	REM	MCC		REM	MCC		REM	MCC		REM	MCC		REM	MCC	
TFID															
ZHU09_Rdr1-11	—	0.800		4	0.800		—	—		—	—		—	—	
ZHU09_Tbs1-9	—	0.776		2	0.793		4	0.793		—	—		—	—	
ZHU09_Gcn4-11	—	0.853		0	0.893		4	0.909		2	0.914		1	0.897	
ZHU09_Yer130c-9	—	0.703		1	0.744		0	0.782		4	0.852		2	0.903	
ZHU09_Phdl	—	0.909		1	0.920		4	0.920		—	—		—	—	
ZHU09_Tye7-11	—	0.685		0	0.742		4	0.807		1	0.863		2	0.874	
ZHU09_Stb3	—	0.823		4	0.823		—	—		—	—		—	—	
ZHU09_Ypr196w-11	—	0.645		1	0.691		2	0.747		4	0.747		—	—	
ZHU09_Nrg1-9	—	0.852		0	0.903		2	0.914		1	0.914		—	—	
ZHU09_Sut2-11	—	0.852		4	0.852		—	—		—	—		—	—	
ZHU09_Fkh2-9	—	0.771		4	0.771		—	—		—	—		—	—	
ZHU09_Ykl222c-9	—	0.764		2	0.791		0	0.820		4	0.891		1	0.920	
ZHU09_Rgt1-11	—	0.715		0	0.751		4	0.789		1	0.898		2	0.903	
ZHU09_Ybr239c	—	0.886		2	0.891		0	0.928		4	0.932		1	0.914	
Continued on next page															

Table D.1 – continued from previous page

TFID	Number of Features Removed													
	0			1			2			3			4	
	REM	MCC		REM	MCC		REM	MCC		REM	MCC		REM	MCC
ZHU09_Srd1-9	—	0.758		0	0.769		4	0.809		1	0.891		2	0.891
ZHU09_Ecm22	—	0.811		2	0.824		4	0.824		—	—		—	—
ZHU09_Skn7	—	0.654		0	0.708		4	0.786		1	0.848		2	0.859
ZHU09_Gal4-9	—	0.731		2	0.758		0	0.765		4	0.840		1	0.920
ZHU09_Ypr196w-9	—	0.620		1	0.679		0	0.763		2	0.797		4	0.880
ZHU09_Mbp1	—	0.936		0	0.974		2	0.969		—	—		—	—
ZHU09_Tbs1-11	—	0.817		0	0.843		4	0.903		1	0.920		2	0.925
ZHU09_Rap1-9	—	0.473		0	0.542		2	0.645		4	0.720		1	0.903
ZHU09_Spt15	—	0.863		1	0.880		2	0.915		0	0.927		4	0.942
ZHU09_Mig1	—	0.850		2	0.862		0	0.868		4	0.888		1	0.909
GB11_Hap1	—	0.730		0	0.744		4	0.834		1	0.875		2	0.904
GB11_Upc2	—	0.881		4	0.881		—	—		—	—		—	—
GB11_Vhr1	—	0.743		0	0.826		4	0.851		1	0.909		2	0.915
GB11_Ste12	—	0.668		0	0.686		4	0.741		1	0.862		2	0.898
Continued on next page														

Table D.1 – continued from previous page

TFID	Number of Features Removed											
	0		1		2		3		4			
	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC	REM	MCC
GB11_Cad1	—	0.571	2	0.645	0	0.645	1	0.782	4	0.862	—	—
GB11_Ybr033w	—	0.633	1	0.691	0	0.741	2	0.819	4	0.914	—	—
GB11_Mot3	—	0.833	0	0.857	2	0.851	—	—	—	—	—	—
GB11_Hmlalpha2	—	0.832	0	0.856	4	0.868	2	0.891	1	0.881	—	—
GB11_Ylr278c	—	0.756	2	0.787	4	0.787	—	—	—	—	—	—
GB11_Nrg2	—	0.504	0	0.524	4	0.596	1	0.794	3	0.174	—	—
GB11_Zap1	—	0.753	1	0.799	0	0.809	4	0.898	—	—	—	—
GB11_Stb5	—	0.740	0	0.789	4	0.782	—	—	—	—	—	—
GB11_Yer184c	—	0.667	2	0.689	0	0.750	4	0.826	—	—	—	—
GB11_Abf1	—	0.589	0	0.616	4	0.694	1	0.800	—	—	—	—
GB11_Cin5	—	0.661	0	0.691	1	0.775	2	0.857	—	—	—	—
GB11_Gcr1	—	0.777	0	0.783	4	0.809	1	0.914	—	—	—	—
GB11_Cst6	—	0.838	4	0.838	—	—	—	—	—	—	—	—
GB11_Stb4	—	0.809	2	0.809	—	—	—	—	—	—	—	—
Continued on next page												

Table D.1 – continued from previous page

TFID	Number of Features Removed							
	0		1		2		3	
	REM	MCC	REM	MCC	REM	MCC	REM	MCC
GB11_Vhr2	—	0.846	3	0.871	2	0.875	4	0.875
GB11_Ecm22	—	0.824	0	0.831	4	0.933	2	0.938
GB11_Sko1	—	0.884	0	0.893	4	0.898	2	0.921
GB11_Sut1	—	0.785	0	0.811	4	0.851	1	0.925
GB11_Yap3	—	0.638	2	0.641	1	0.654	0	0.874
GB11_Hac1	—	0.808	0	0.821	4	0.850	1	0.886
GB11_Stp1	—	0.867	0	0.937	2	0.953	4	0.953
GB11_Pdr3	—	0.870	1	0.870	—	—	—	—
GB11_Msn1	—	0.849	2	0.859	0	0.883	4	0.920
AVERAGE	0.766		0.795		0.820		0.853	0.871

REM: Feature that improved the model performance the most when removed in current iteration.