# Predicting Functional Transcription Factor Binding Sites in Human Using Interspecies Comparison

Jimmy Hsin-Chia Chao

School of Computer Science

McGill University, Montreal

July 2015

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Masters of Computer Science

© Jimmy Hsin-Chia Chao, 2015

# Table of Contents

# Table of Figures

# Abstract

Transcription Factor Binding Sites (TFBS) are regions in the genome where Transcription Factor (TF) proteins bind in order to regulate the rate of transcription of one or more nearby genes. As TF plays an important role in gene regulation, much research has been directed at understanding the behavior and mechanism of these proteins in the hopes of gaining a better understand of the gene regulatory network in cells. A part of this on-going research is the discovery of TFBS. Protein-DNA interaction experiments, such as ChIP-Seq can accurately identify locations of TFBS on the genome *in vivo*. However, one major drawback of experimental methods is its time and cost.

Computational methods have been proposed that finds TFBS by scanning the genome looking for matches to the sequence preference of TF; but this method faces the issue of high false positive rate. In our research, we develop a new method that filters out these false positive predictions by training a Support Vector Machine (SVM) classifier that learns whether a computationally inferred TFBS is biologically functional based on inter-species conservation and the presence of other TF. Using the genome of 35 species as well as the inferred genome of their ancestors and the data of 38 different TF, we were able to build a classifier that could predict with up to 90% accuracy whether a computationally predicted TFBS is biologically functional for many of the TF we investigated.

# Résumé

Les sites de liaison de facteurs de transcription (SLFT) sont des régions dans le génome où des proteins appellées facteurs de transcription (FT) se lient à l'AND afin de réguler le taux de transcription d'un ou plusieurs gènes voisins. Comme les FT jouent un rôle important dans la régulation des gènes, beaucoup de recherche a été faite pour comprendre le comportement et le mécanisme de ces protéines dans l'espoir de mieux comprendre le réseau de régulation génique dans les cellules. Une partie de cette recherche en cours est la découverte de SLFT. Des expériences d'interaction protéine-ADN, telles que ChIP-Seq peuvent identifier avec précision l'emplacement des TFBS sur le génome in vivo. Cependant, un inconvénient majeur de ces méthodes expérimentales est leur coût en terme de temps et d'argent.

Des méthodes de calcul ont été proposées pour identifier les SLFT en scannant le génome à la recherche de correspondence au motif reconnu par le FT, mais ces méthodes ont un taux élevé de faux positifs élevé. Dans notre recherche, nous développons une nouvelle méthode qui filtre ces prédictions par la formation d'un classificateur Support Vector Machine (SVM) qui apprend si un SLFT candidat est biologiquement fonctionnelle basée sur la conservation inter-espèces et la présence d'autres SLFT. En utilisant le génome de 35 espèces ainsi que le génome présumées de leurs ancêtres et les données de 38 TF différents, nous sommes en mesure de construire un classificateur qui peut prédire avec jusqu'à 90% de précision si un SLFT candidate est biologiquement fonctionnel, pour un bon nombre de TF considérés.

# Acknowledgements

As many graduate students would agree, writing the thesis can be a long arduous journey. Without the countless support and encouragement from so many wonderful individuals, this journey would not have been possible for me.

First and foremost, I would like to give my deepest and sincere gratitude to my supervisor, Professor Mathieu Blanchette, whose wisdom, patience and steadfast generosity has been a guiding lamppost in the writing of my thesis every step of the way.

I would also like to thank my external reviewer, Professor Jérôme Waldispühl, for taking the time to read my thesis and provide helpful comments.

In addition, my two years of master's program would not have been as enjoyable as it was without all my friends and colleagues who have been some of the most kind and light-hearted people I have met. I am especially grateful for my parents and my brothers for always being there for me in my time of need. Finally, I want to thank McGill, its staff and my graduate coordinator for making all this possible.

# Chapter 1: Introduction

Since the discovery of DNA, biologists have been puzzled by a phenomenon known as the "C-Value Paradox" (coined by C.A. Thomas 1971). This paradox was reiterated in 2006 by Van Straalen and Roelofs as a "remarkable lack of correspondence between genome size, number of protein-coding genes and organism complexity, especially among eukaryotes" (van Straalen & Roelofs, 2006). It is clear that a more sophisticated mechanism of regulation of gene expression must exist within cells. One of these regulation mechanisms is the involvement of Transcription Factors (TF), which are proteins that bind to specific DNA sequences, or sequence motifs, on the genome and can act as catalysts for transcription. The locations where binding occurs are called Transcription Factor Binding Sites (TFBS); they are typically located in the promoter or enhancer regions of the gene they regulate. TF regulate genes that are spatially close to where they bind and mutations in the binding site can interfere with the rate at which these genes are normally expressed and lead to alterations in phenotypic characteristics (Kamanu et al., 2012) resulting in a variety of diseases (Baldwin, 2001). Thus, there is a growing interest in the identification of TFBS locations as it not only helps us paint a better picture of how our gene regulatory network works, but also in the study of genetic disorders (Kamanu et al., 2012).

Experimental methods, such as ChIP-Seq (discussed later in this chapter), provide a relatively low cost way of accurately identifying TFBS locations on the genome (Park, 2009). However, ChIP-Seq experiments require large amounts of starting material (e.g. cells and tissues) and the procedure is time consuming (several days to weeks) (Epigentek, 2015). Given that the experiment must be repeated for different TF, experimental methods can quickly become expensive and impractical on a larger scale. Cost efficient computational methods have also been suggested, which will be discussed later in this chapter, but face the issue of false positive predictions. The motivation of this research is to find an effective computational method that predicts TFBS while reducing the false positive prediction rate by using the

knowledge that functional TFBS tends to form clusters, are under selective pressure and tend to be conserved between related species (Loots & Ovcharenko, 2004).

We begin this chapter by reviewing the biology of TF and TFBS in relation to transcriptional regulation, followed by a survey of current experimental and computational methods of discovering functional TFBS on the genome. Furthermore, we will explore topics of machine learning and their application to transcription factor binding site prediction.

## 1.1 Transcriptional Regulation

The expression of genes, from DNA to protein, begins with the process of transcription. During transcription, genes are copied from DNA to messenger RNA (mRNA) by the enzyme RNA polymerase as shown in Figure 1. mRNA then exits the nucleus and are later translated into protein (Alberts, 2007). The production of proteins is strictly regulated in cells and its rate may increase, or decrease, depending on various factors such as environmental changes or mutations (Lobo, 2008).

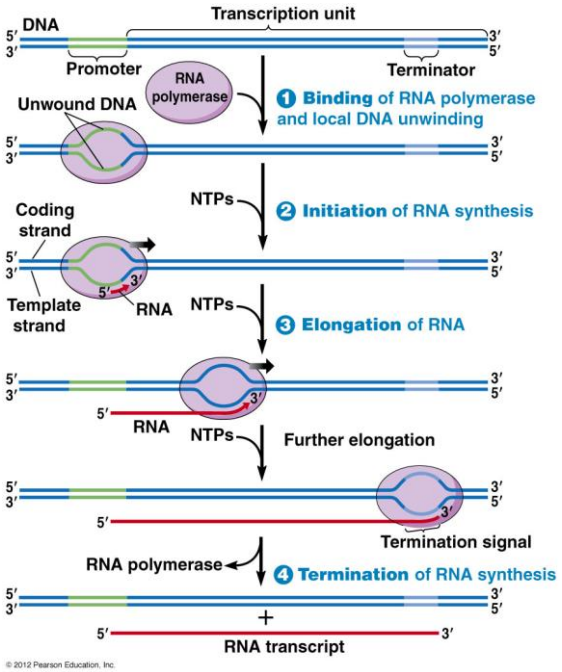



**Figure 1. Process of the Transcription of mRNA, reproduced from (Jeff Hardin, 2012).**

Gene expression is primarily regulated on two levels:

- Transcription: This is first step in gene expression where segments of DNA are copied into strands of messenger RNA (mRNA) by the RNA polymerase enzyme. In eukaryotes, binding of RNA polymerase is enabled by prior binding of TF on promoters, which are regions approximately 30-100 bps upstream from a gene's transcription start site (TSS) (Smith, Marks, Lieberman, & Marks, 2005). Additional proteins and TF co-factors then bind to the TF complex on the promoter to control the rate of transcription. TF also binds to regions on the genome known as enhancers, which are located sequentially far from genes relative to promoters (>10kpbs). Enhancers can become spatially close to genes after DNA folding allowing bound TF to regulate the genes nearby (NatureEducation, 2014a). Modifications to the structure of chromatin, such as DNA folding and methylation, can also affect transcription by preventing the binding of TF (Phillips, 2008).
- Post-Transcription: Successfully transcribed mRNAs are then processed prior to being translated to specific proteins. Before exiting the nucleus, mRNAs are spliced by spliceosome where introns are removed and exons are joined. Alternative splicing allows exons to be joined in different ways such that one gene can code for multiple proteins (Carmody & Wente, 2009). A poly(A) tail is added to the mRNA at the 3' end in order to increase its stability and half-life. Long poly(A) tail also encourages translation as it recruits binding of ribosomes to the mRNA (Preiss, 2000). Processed mRNA must then be exported from the nucleus to the cytoplasm by assembling into complicated ribonucleoprotein, recruiting exporter proteins and attaching themselves to nuclear pore complexes (Kohler & Hurt, 2007); all of which can be regulated by various signal transduction pathways (Quaresma, Sievert, & Nickerson, 2013). Once in the cytoplasm, translation can be interrupted by targeted degradation of mRNA or by preventing the ribosome from accessing the translation start codon (Day & Tuite, 1998), among others.

This complex regulation of gene expression gives cells the ability to differentiate into specialized cells despite having the same DNA. It also allows cells to carry out different cellular functions and alter the rate of protein production when subjected to environmental stimuli. In most eukaryotes, genes are usually turned "off" by default and are turned "on" by TF (this is often the reverse for bacteria) (Babu, Luscombe, Aravind, Gerstein, & Teichmann, 2004). This occurs when TF binds to the promoter region of a gene to "activate" it. To date, less is known about post-transcription gene regulation compared to transcriptional regulation due to the intricate network of control mechanisms involved (Besse & Ephrussi, 2008). In this section, we will focus on the understanding of the biology of gene regulation at the transcriptional level as it relates directly to our research.

### 1.1.1 Transcription Factors

TF are proteins that bind to specific DNA sequence patterns to control the rate in which genes are transcribed. They can be found in all living organisms. In humans, approximately a tenth of all protein-coding genes are for TF proteins, making it the largest of all protein classes (Babu et al., 2004). TF tend to work in groups, or complexes, by establishing multiple interactions with each other allowing for varying degrees of control over the rate of transcription (Thomas & Chiang, 2006) as seen in Figure 3. The structure of TF typically contains the followed domains (Latchman, 1997):

- A DNA-Binding Domain (DBD), which binds to specific DNA sequences.
- A Trans-Activation Domain (TAD), which allows interactions with other co-regulator proteins.
- A Signal Sensing Domain (SSD), which senses external signals and propagates them to the rest of the TF complex to increase or decrease transcription (not always present).

#### *1.1.1.1 DNA Binding Domain*

Different TF have different structures and different DBD. The Sequence preference of a TF DBD is often not strict, but follows a probabilistic distribution over nucleotide sequences. This sequence pattern

preference, or motif, is generally represented using a Position Weight Matrix (PWM), which we will discuss in section 1.3.1.

#### *1.1.1.2 Trans-Activation Domain*

The TAD also varies between TFs; they provide binding sites for co-activators (protein that increase transcription) or co-repressor (protein that decrease transcription), which work in conjunction with the TF to regulate the rate of transcription. Some TF have multiple TADs, for example, TF SP1 contains two glutamine-rich TADs that interacts with TAFII (Frietze & Farnham, 2011).

#### *1.1.1.3 Signal Sensing Domain*

This region is not present in all TF. The SSD is the part of the TF that "senses" or binds to external chemical signals (e.g. steroid hormones or cAMP) and transmits the signal to the rest of the TF complex to speed up or slow down transcription. It is possible for SSD to reside on other proteins that are part of the transcription complex (e.g. a complex of multiple TF and other proteins). For example, a protein with SSD can bind to the TAD region of the TF giving it the ability to indirectly sense chemical signals.

### 1.1.2 Transcription Factor Binding Sites

Transcription Factor Binding Sites (TFBS) are locations on the genome where TF binding occurs. All sub-sequence on the genome that matches a TF's binding motif are possible binding sites; however, DNA is organized such that sequence strands are compacted by histones into nucleosomes and are often not accessible to TF (Beato & Eisfeld, 1997). To make TFBS within nucleosome accessible, cells can use chromatin remodelers to shift or remove nucleosome. Thermal input could also cause unwrapping of DNA strand from nucleosome and allow temporary access to TFBS.

Since the binding preference of TF is not fixed, but follows a pattern, the nucleotide sequences of TFBS for the same TF can differ between sites. This pattern is usually represented by a Position Weight Matrix. Figure 2 shows the PWM of two different TF motifs. We notice that while some positions have a

strong preference for one nucleotide over another, other positions could have equal preference for multiple different nucleotides at once. For example, E2F-1 shows preference for TGCGC between position 5-9 and equal preference of “A”, “G”, or “T” for position 4. Also, TFBS can vary in length. Longer binding sites can have better specificity, but are less robust to mutations (Stewart, Hannenhalli, & Plotkin, 2012). The typical length is between 5-31 nucleotides with an average length of 10 nucleotides.

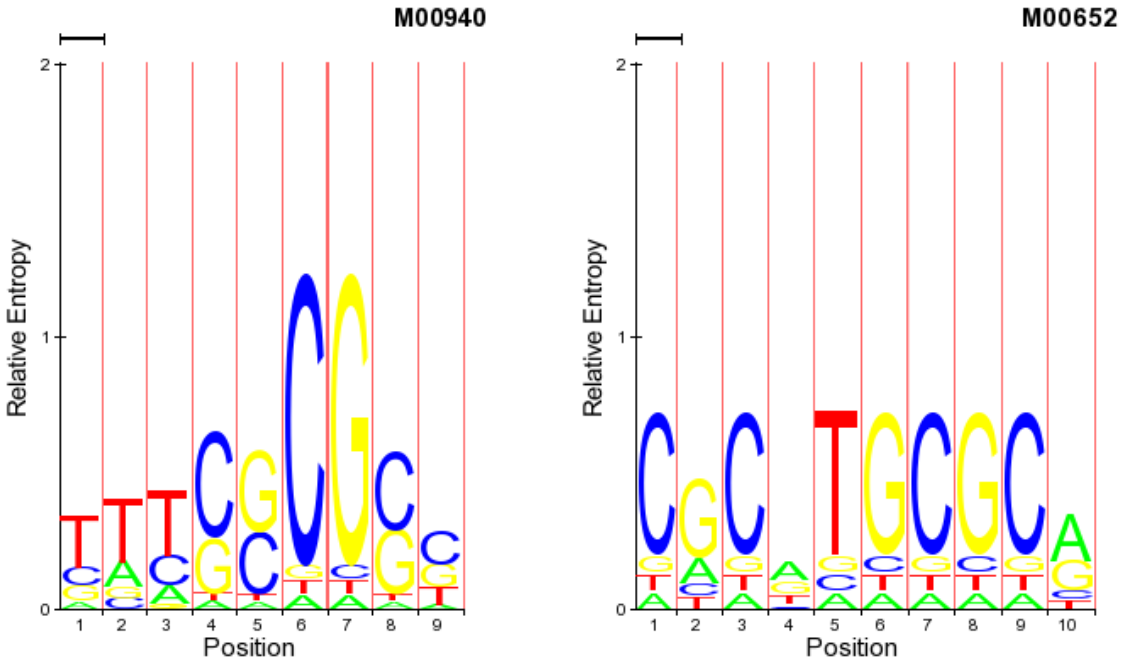



**Figure 2. PWM for NRF-1 and E2F-1. Nuclear Respiratory Factor 1 regulates cellular growth and nuclear genes for respiration (right) and E2F-1 family is involved in the control of cell cycle in human (left) (Marinescu, Kohane, & Riva, 2005).**

TFBS are generally located relatively close to genes that need to be activated by the TF. In particular, TFBS are sometimes found on the promoter of the gene, which are regions of 100-1000 bps in length located near the transcription start site of genes. RNA polymerase will bind to the promoter to initiate transcription and binding of TF on the promoter can help in the recruitment of RNA polymerase (NatureEducation, 2014b). TFBS are also found in enhancers, which are regions 50-1500 bps in length often located quite far (10-1000 kbps) away from genes. Binding of TF on enhancers can assist in the transcription of genes that are far away in terms of base pairs due to DNA looping (NatureEducation,

2014a). A good interpretation of this is shown in Figure 3 where we see the action of multiple TF on areas of the genome that are not sequentially adjacent to each other but are only close after a three dimensional folding.

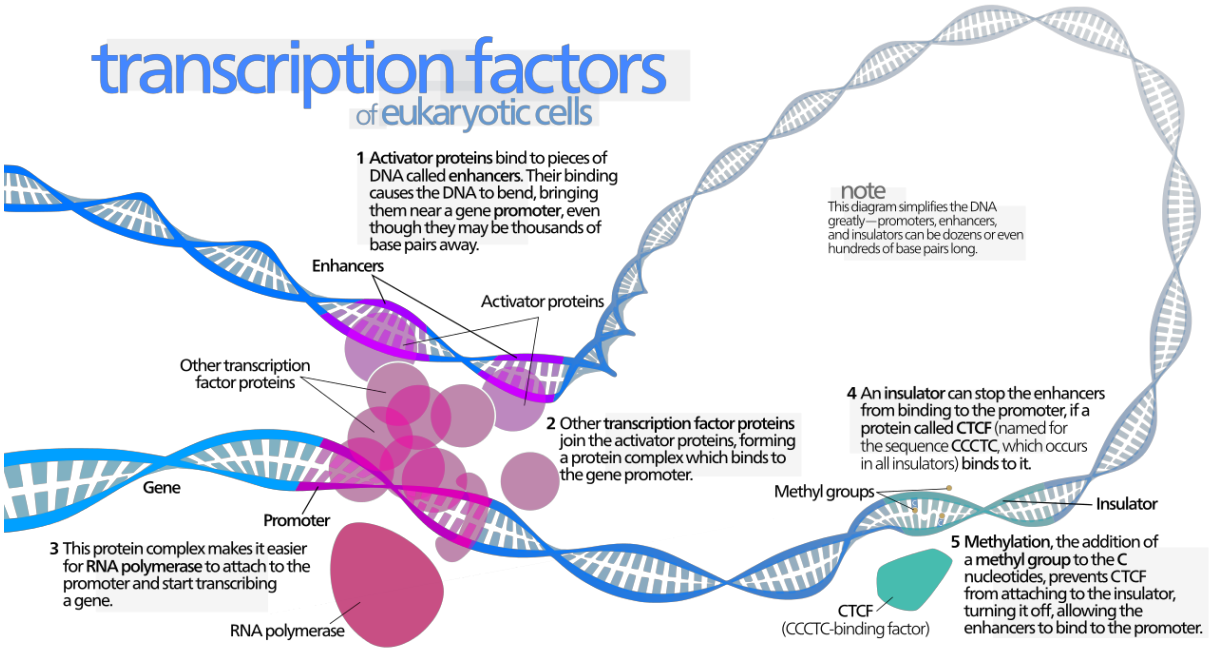



**Figure 3. Illustration of the action of TF on gene regulation (Song, 2012).**

### 1.1.3 Evolutionary Conservation and TFBS Turnover

Functional TFBS are often conserved throughout evolution and highly conserved binding sites are observed through interspecies comparison. These highly conserved sites are expected to have a more radical effect on gene expression and mutations in these sites tend to be associated with diseases (Claussnitzer et al., 2014). Nevertheless, many TFBS locations that are functional in one species may not be functional or conserved in other species even if both species need to express the same genes. For example, experiment shows that 32% to 40% of functional sites in human are not functional in rodents (Dermitzakis & Clark, 2002). This may be due to turnover where a new site is created close to an existing one such that the new site is used while old sites are intact but not used (Dermitzakis & Clark, 2002).

The relatively short length of TFBS and low specificity of binding often result in numerous regions in the genome having near matches to the motif. Sometimes, a single point mutation is sufficient to create a new binding site. As a result, new sites may relax the evolutionary need for TFBS conservation since old sites may be removed without serious phenotypic consequences (Dermitzakis & Clark, 2002). Different binding sites have different rates of turnover; for example, binding sites with longer motif have a lower chance of turnover since they are less likely to be created by point mutations. This behavior of TFBS turnover (creation or deletion of sites) has been observed in mammals and makes it difficult to identify which sites are functional but have undergone turnover in other species such that it appears the site is not evolutionary conserved.

## 1.2 Experimental Approaches to TFBS Identification

Several experimental methods exist to gather data regarding the identification of both the sequence motif of a TF and their binding locations on the genome. They are particularly important in that the data generated are considered "ground truth" used as benchmark for computational TFBS prediction algorithms. Two fundamental types of approaches are in vitro and in vivo studies, which we discuss below.

### 1.2.1 In vitro Studies

*In vitro* methods study biological interactions in controlled laboratory environments (e.g. a test tube) rather than in their normal biological context (i.e. within cells). TFBS studies done *in vitro* are often related to the discovery of binding site sequence motif and identification of binding properties such as the binding energy landscape (Maerkl & Quake, 2007) or biophysical parameters governing binding events (Geertz & Maerkl, 2010; Maerkl & Quake, 2009). In this section, we will look at two *in vitro* methods of TFBS motif discovery known as SELEX and PBM.

#### *1.2.1.1 Systematic Evolution of Ligands by Exponential Enrichment (SELEX)*

One of the earliest *in vitro* methods for discovering TF binding sites is the Systematic Evolution of Ligands by Exponential Enrichment (SELEX) (Djordjevic, 2007). First, solutions with random DNA oligonucleotides are mixed with TFs. Oligonucleotides with bounded TFs are then amplified by PCR. This procedure is repeated with oligonucleotides that showed high likelihood of TF binding in the previous round until we have a solution that contains only high affinity binding oligonucleotides (oligonucleotides that are almost always bound by TF when mixed in solution). These oligonucleotides are then sequenced and the binding consensus sequence is derived. The drawback is that only high affinity sequences are discovered, which does not model the irregularity of TF binding in living cells (e.g. binding may not occur in live cells although having a motif match). Recent advances in sequencing techniques remove the need to repeat rounds of SELEX binding and amplifying. Instead, a single round using the whole genome sequence is sufficient to capture the high affinity binding sites as well as the non-linearity of binding motifs (Zykovich, Korf, & Segal, 2009), as shown in Figure 4. However, a sufficiently high number of TF copies needs to be produced such that enough binding occurs in a single round, which can be expensive.

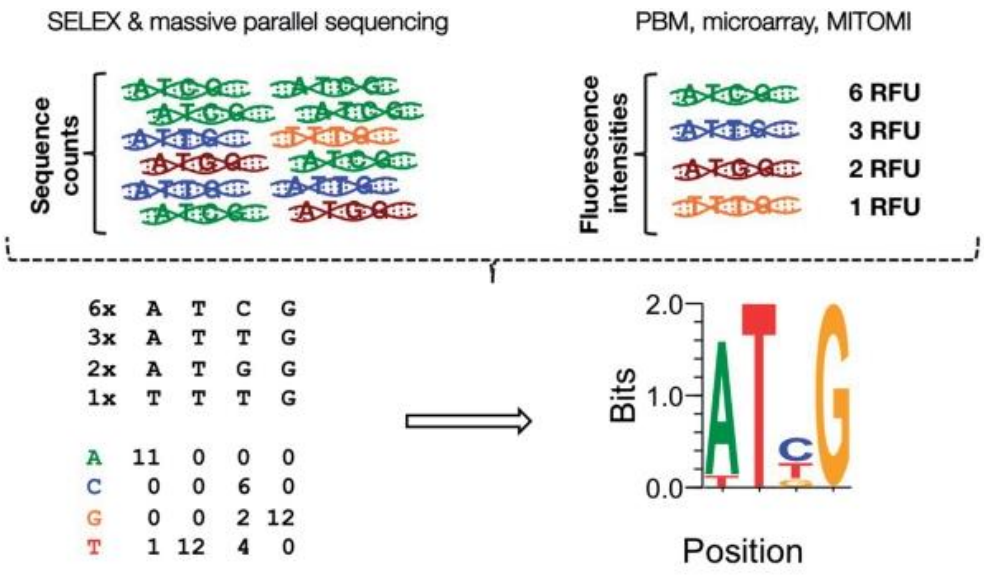



**Figure 4. Example of SELEX-Seq where TF-bound DNA fragments are enriched and the frequencies of nucleotides at each position are counted to create a PWM (Geertz & Maerkl, 2010).**

#### *1.2.1.2 Protein-Binding Microarrays (PBM)*

DNA microarrays is a technology used to perform experiments on multiple strands of DNA (from tens to millions depending on experiment) simultaneously (Trevino, Falciani, & Barrera-Saldana, 2007). A flat surface (usually glass) is divided evenly into rows and columns forming a two dimensional matrix. Each cell within this matrix has multiple identical strands of DNA attached to it. Different cells contain different DNA sequences. Solution containing agents that would react with these DNA sequences are added to the surface and reaction is observed under a microscope. PBM make use of this technology by injecting the TF of interest within each of the microarray cells and allowing time for the TF to bind to the DNA strands. The microarray is then washed such that all TF not bound to any DNA strands are removed. Protein-specific fluorophore-coupled antibodies are then added to each cell such that they attach themselves to binding TF as shown in Figure 5. Since the DNA sequence of each cell is known, any cell that glows under fluorescent lighting indicates a successful binding. Bound sequences are recorded and typically transformed to a PWM (Geertz & Maerkl, 2010).

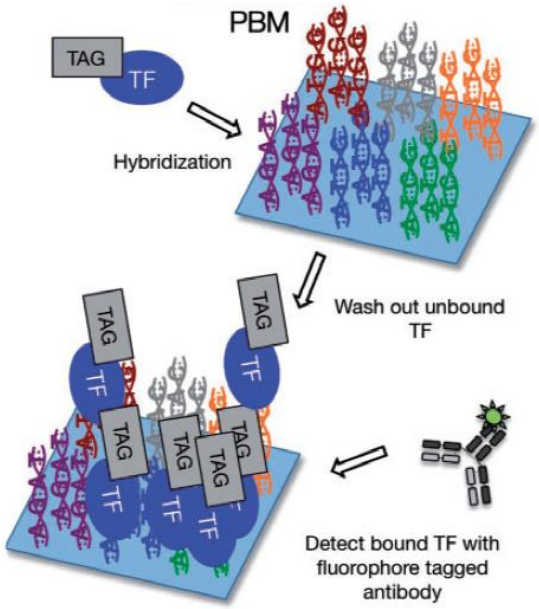



**Figure 5. PBM experimentation steps (Geertz & Maerkl, 2010).**

### 1.2.2 In vivo Experiments

*In vivo* methods study biological components in a living organism rather than in a test tube. TFBS studies done *in vivo* are often related to the biological context of binding interactions. Common methods used in this category to detect TF-DNA interactions include Chromatin Immunoprecipitation (ChIP), which is generally followed by either hybridization to a micro-array (ChIP-Chip) (Buck & Lieb, 2004) or massively parallel sequencing (ChIP-Seq) (Landt et al., 2012).

#### *1.2.2.1 Chromatin Immunoprecipitation (ChIP)*

Chromatin immunoprecipitation is a widely used technique in the study of protein-DNA interaction in cells. It aims to identify active binding sites on the genome *in vivo*, meaning that the DNA sequences under investigation are in their living state in the cell. Unlike *in vitro* experiments where multiple artificially amplified DNA strands are used, ChIP analyzes the entire genome as a whole. As a result, inaccessible sites, such as those bound by histone, will not have any TF-DNA interactions and are reflected in the results. The steps for ChIP (Figure 6) are as follows (Carey, Peterson, & Smale, 2009):

1. Protein (TF) and DNA are crosslinked *in vivo.*
2. DNA is then sheared into fragments of ~500-700bp by sonication (Sambrook & Russell, 2006).
3. DNA fragments that are bound by the TF are selectively immunoprecipitated with protein-specific antibodies (Bonifacino, Dell'Angelica, & Springer, 2001).
4. Precipitated TF-DNA complex is then purified and identified by either DNA microarray or massively parallel DNA sequencing.

#### *1.2.2.2 Chromatin Immunoprecipitation with DNA Microarray (ChIP-chip)*

Once TF-bound DNA fragments are precipitated, DNA microarray can be used to identify these regions. First, the TF-DNA complexes is separated (typically by heat) and purified such that only single stranded DNA remains. Each strand is then tagged with fluorescent material. A DNA microarray is then prepared where each cell contains short known single stranded DNA oligos. The fluorescent tagged DNA strands is

then poured over the microarray such that they can find their complementary strand that is attached to the microarray in one of the cells and hybridize forming a double stranded DNA.

After a sufficiently long duration of time to allow for hybridization, the microarray is put under a microscope and fluorescently lit to see which cells are glowing, revealing the sequences that were bound by the TF. Bioinformatics methods are then used to piece the sequences back together to determine which areas on the genome corresponds to the bound DNA strands.

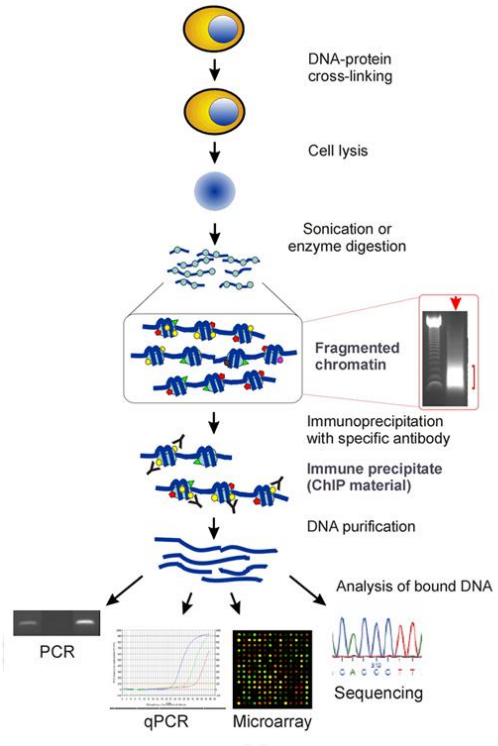



**Figure 6. Workflow of ChIP process. After purification, sequencing is done using various techniques such as PCR, qPCR, microarray or next generation sequencing (Carey et al., 2009).**

Although ChIP-chip is a powerful technique that provides high-resolution genome-wide maps, it is not without limitations. Firstly, ChIP-chip experiments are expensive as DNA microarrays are costly. In most cases, experiments are repeated at least three times to obtain biologically meaningful mapping of the binding sites. Secondly, due to the limited number of cells in the microarray, the resolution of sequencing is generally capped at 30-100 bps. Moreover, only binding site that matches one of the

sequence strands on the microarray will be detected. Having enough probes on a microarray to cover the whole human genome is too expensive. High density microarray can be used to provide greater combinations of DNA sequences, however this will introduce noise as hybridization is highly complex and cross-hybridization between imperfectly matched sequences occur frequently (Park, 2009).

#### *1.2.2.3 Chromatin Immunoprecipitation with Massively Parallel DNA Sequencing (ChIP-Seq)*

Recent advances in next generation sequencing tools such as Illumina and Ion Torrent have greatly reduced the cost of sequencing such that ChIP-Seq have now replaced ChIP-chip as the most common technique to study protein-DNA interactions. ChIP-Seq uses massive parallel sequencing techniques to sequence the precipitated TF-DNA complexes and can achieve a maximum resolution of individual nucleotides (Park, 2009). The technology behind how sequencing is performed will not be discussed here as different company uses different proprietary techniques.

ChIP-Seq offers advantages over ChIP-chip in that the detectable combination of nucleotide sequence is not limited to the repertoire available on the microarray, thus greatly increases its resolution. Repeated sequences that are usually masked out on microarrays can be sequenced using ChIP-Seq. This is especially useful in studying regions involving microsatellites or long terminal repeats. ChIP-Seq also does not suffer from noise in hybridization the way ChIP-chip does. Currently, although the cost of sequencing has been steadily decreasing in recent years, it is not significantly cheaper than the use of microarrays.

## 1.3 Computational Approaches to TFBS Prediction

In this section, we discuss the current progress in the field of TFBS prediction using computational methods. We begin by reviewing the position weight matrix (PWM) model, a commonly used statistical representation of TF binding motif, followed by an analysis of various existing techniques.

### 1.3.1 Position Weight Matrix

Position Weight Matrices, also known as position specific weight matrix, are used to represent motifs in biological sequences by representing the base frequency at each position. They are widely used to characterize binding sites for TF. TFBS PWM models are learned from experimental data and this process can be as straightforward as counting the frequencies of nucleotide that is seen at each position. Usually, experimental data will first be used to produce a position frequency matrix (PFM) as shown in

Figure 7 where the DNA sequence that results in active binding are collected and frequency of nucleotide appearing in a particular position are recorded.

| Position Frequency Matrix | | **Position** | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| **Nucleotide** | **A** | 1 | 3 | 2 | 1 | 1 | 1 | 1 | 1 | 1 |
| | **C** | 2 | 1 | 2 | 4 | 4 | 7 | 1 | 4 | 3 |
| | **T** | 5 | 4 | 5 | 1 | 1 | 1 | 1 | 1 | 3 |
| | **G** | 2 | 2 | 1 | 4 | 4 | 1 | 7 | 4 | 3 |

**Figure 7. PFM for NRF-1. The columns are the different positions in the binding sequence and the rows are the different nucleotides. Each entry in the matrix represents the number of observed binding sequence where a particular nucleotide is seen at the indicated position.**

Once we have the PFM, the following formula can be used to calculate the log-likelihood ratio value of each nucleotide in each position:

$$W_{b,i} = \log_2 \frac{p(b,i)}{p(b)}, \qquad p(b,i) = \frac{f_{b,i} + s(b)}{N + \sum s(b')},$$

where $W_{b,i}$ is the PWM value of nucleotide $b$ in position $i$; $p(b)$ is the background probability of base $b$, which is usually set to the genome wide frequency of base $b$. It is possible for the frequency of one type of nucleotide to be much greater than another, such as in a CpG island, where the background frequency of 'C' and 'G' is significantly higher than that of 'A' and 'T'; $p(b,i)$, sometimes called position probability matrix, is the probability of base $b$ in position $i$; $f_{b,i}$ is the count of base $b$ in position $i$; $N$ is

the sum of the sites per position (usually the number of identified binding sites); $s(b)$ is the pseudo-count function which is used for small datasets to address issues with zero in the denominator (Tang, 2013). PWMs assume independence of nucleotides at different positions, meaning that the presence of a nucleotide in one position does not impact the probability of a particular nucleotide showing up at another position.

#### *1.3.1.1 Finding TFBS using PWM*

Once a PWM is derived, it can be used to scan any DNA sequence to find potential matches. A common method is to run a window of size equal to the length of the matrix along the sequence and at each position determine a score by summing the PWM values corresponding to the nucleotide at each position:

$$Score_{pos} = \sum_{i=1}^{len} PWM(S_{pos+i}, i)$$

where $Score_{pos}$ is the score at position $pos$ in the sequence; $len$ is the length of the PWM, $S_k$ is the nucleotide at position $k$ in the sequence, $PWM(n, m)$ is the PWM information value for nucleotide $n$ at position $m$ in the PWM. The result will be a list of scores for every position in the DNA sequence. A threshold can then be used to filter out the ones with lower identity score. There are many tools available that perform PWM matching such as MOODS (Korhonen, Martinmaki, Pizzi, Rastas, & Ukkonen, 2009) and MatchTM (Kel et al., 2003), which uses specialized algorithms optimized for speed.

#### *1.3.1.2 Optimization of PWM*

PWMs are statistical representation of experimental data that tells us, at each position, how much information content can be gained by observing a particular nucleotide given a background nucleotide frequency. However, like all statistical analysis, the choice of data and assumptions can often lead to bias in the PWM as well as other statistical pitfalls (Chatfield, 1991). The use of a constant background

nucleotide frequency can be problematic, as nucleotide frequencies are not equally distributed across the genome, which can lead to bias in the PWM. Inaccuracies in PWM can also arise due to technological short comings, such as the limit is sequencing resolution of Protein Binding Microarray (discussed earlier in section 1.2.2.2) (Orenstein, Linhart, & Shamir, 2012). Moreover, not all binding sequences are of the same length, TF binding could skip nucleotides and without knowing which position is skipped the data used to generate the PWM may be inaccurate (Siddharthan, 2010). Much research has been conducted to improve the accuracy of PWMs. We discuss here two approaches that have been proposed to address these issues.

The Genetic Algorithm Method for Optimizing a Position Weight Matrix (GAPWM) (Li, Liang, & Bass, 2007) uses an existing PWM, a set of ChIP sequences and a set of background frequencies to improve the PWM by using a genetic algorithm (Goldberg, 1989). The motivation is to realize that positions within motifs are not necessarily independent. For example, if a nucleotide is strongly dependent on the nucleotides before them, then its occurrence of has low information content in itself. Thus, deriving a PWM by analyzing the observed nucleotides in every position independently may not be as accurate as looking at the motif as a whole. Given a list of experimentally determined binding site sequences, another list of non-binding site sequences of the same length is randomly generated. The strategy is to iteratively alter the PWM such that the accuracy of finding the correct binding sites is maximized. Alteration is done by randomly selecting a combination of PWM position and nucleotide and changing its observed frequency by a small value (±0.02). The sampling space to find the maximum accuracy is quite large; however, using the existing PWM as a reference, the closest local maxima can be found while drastically reducing the sampling space. Since the PWM's accuracy is maximized, the resulting PWM will generally be an improvement or equal in performance to the original PWM.

DIScriminative PAttern Refinement for Position Weight Matrix (DISPARE) is another optimization technique that also iteratively refines existing PWM using ChIP sequences. Similar to GAPWM, a set of

experimentally determined binding sequences are mixed with randomly generated non-binding sequences. All of these sequences are scored against the existing PWM and a threshold value is calculated that would most optimally separate the binding sequence from non-binding sequence based on their scores. All binding sequences with a score above the threshold are then used to generate a new PWM. This is repeated until the Kullback-Leibler Divergence (Kullback & Leibler, 1951) value between the new PWM and previous PWM is below a certain epsilon or the user-defined maximum number of iterations is reached. DISPARE also optimizes the length and phase (shift) of PWM by computing all possible ways of adjusting the size of the motif and shifting the nucleotides to the left and right until there is no significant gain in information content (da Piedade, Tang, & Elemento, 2009).

Both techniques have shown improvements in terms of area under receiver operating characteristic curve (discussed in detail in section 2.1.8.4).

#### *1.3.1.3 Hidden Markov Model PWM*

While PWM assume that nucleotides are independent between positions, a generalization of PWM called the Hidden Markov Model Position Weight Matrix (HMM-PWM) used by the TFBS discovery tool "CONTRASIF" (section 1.3.2.1.2) is an improvement to the original PWM where it takes into account how often pairs of nucleotide tend to appear together. Given a PFM or simply a list of TF's binding sequences (such that a PFM can be generated), an additional co-occurrence matrix is calculated at each position that determines how likely a particular nucleotide is to follow another nucleotide. A HMM is then built where the emission probabilities each state (position) are calculated based on the PFM and the transition probability matrix is built using the co-occurrence matrix. An optimal similarity threshold is then determined and every sub-sequence with an HMM similarity score greater than the threshold is considered a candidate binding site.

#### *1.3.1.4 Transcription Factor Flexible Model*

Another improvement on the traditional PWM is the Transcription Factor Flexible Model (TFFM) introduced by Mathelier et al. (Mathelier & Wasserman, 2013). Noticing that PWM makes the strong assumption of nucleotide independence and does not take into account its variable length, the TFFM uses a Hidden Markov model approach to model the dependence of observing a nucleotide based on the emission of the nucleotide in the position before. The result is a more comprehensive matrix as shown in Figure 8.

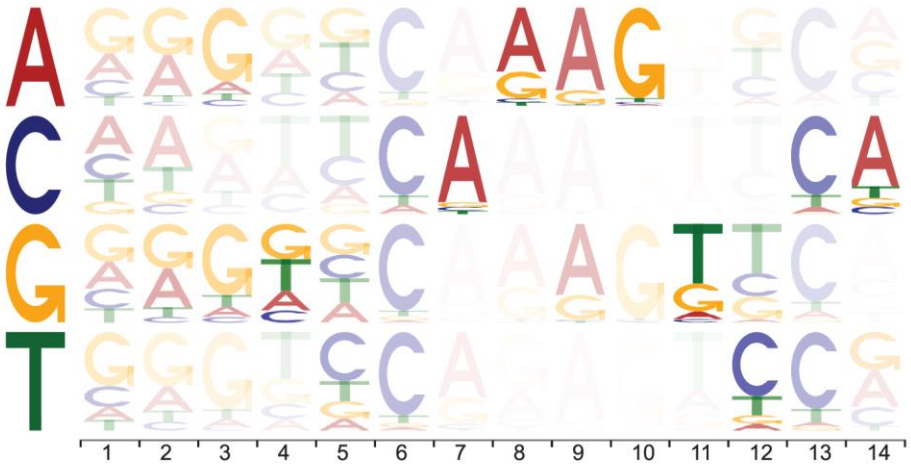



**Figure 8. Illustration of a Transcription Factor Flexible Model for HNF4A. Each column corresponds to a position in the TF's motif. Each row corresponds to the nucleotide seen in the position before. For example, for row 'G' and column 3, the size of the nucleotide is the probability of emitting that nucleotide given the nucleotide emitted in the previous position is a 'G'. The opacity signifies the probability of reaching that nucleotide. (Mathelier & Wasserman, 2013)**

#### *1.3.1.5 Databases of PWMs*

Currently, numerous databases exist that contains experimentally determined PWM data for TF found not only in humans, but in other species as well. Among the largest of these databases are TRANScription FACtor (TRANSFAC) (Matys et al., 2003) and JASPAR (Sandelin, Alkema, Engstrom, Wasserman, & Lenhard, 2004). Both are available online and manually curated. TF data are categorized into families and classes based on the features of their binding domain. Although both databases

contain PWM data for many TF, TRANSFAC have a larger collection (>5500) compared to JASPAR (~590 non-redundant). The trade-off of using the larger TRANSFAC database is its subscription fee, whereas JASPAR is free and open-sourced. TRANSFAC also offers its older database for free but it is extremely outdated. Both databases are updated regularly with improved or new binding PWM.

### 1.3.2 Comparative Genomics Approaches

Although a complete sequencing of an organism's DNA contains all the information required to encode that organism, the sequence alone tells us little about how the genetic information leads to observable phenotypes. The main principle of comparative genomics is to look for similarities between genomes of different species recognizing that important functional regions of the genome tend to experience positive selective pressure (Hardison, 2003). In this section, we look at a number of comparative genomics approaches to TFBS prediction that either looks directly at orthologous genomic regions of various species or performs a multiple alignment of multiple species.

#### *1.3.2.1 Without the Multiple Sequence Alignment of Species*

##### 1.3.2.1.1 Phylogenetic Footprinting

Phylogenetic footprinting is a technique used to identify the location of TFBS in non-coding regions without prior knowledge about TF binding motifs. This technique makes use of orthologous portions of the genome in multiple species to find regions of sequence conservation. It uses the fact that functional genomic regions tend to be well conserved in diverse species and regions of DNA that are essential for gene expression will experience stronger selective pressure compared to less critical areas (Ganley & Kobayashi, 2007). In this type of studies, it is important to select species that are similar enough to find homology yet diverse enough such that alignment represents more than simply species similarity, as illustrated in Figure 9.

The protocol of phylogenetic footprinting is as follows:

1. Identify the gene of interest.
2. Choose species with orthologous genes.
3. Perform global alignment of the promoter/enhancer regions of the genes.
4. Identify regions of elevated conservation.

The advantage of this method is that functional binding sites can be inferred using only data from sequenced species without prior knowledge of TF PWM. However, disadvantages include the inability to identify binding sites that are 1) only functional in a few species, 2) still functional even after mutations due to the low specificity binding of some TF or 3) have undergone TFBS turnover.

It is important to realize that not all conserved regions are functional as they may be conserved by chance. A region is only identified as functional when the mutation rate of the nucleotides within it is statistically lower than that of the surrounding non-functional regions. Numerous methods have been developed to address this issue such as the consensus building greedy algorithm (Stormo & Hartzell, 1989) that finds a multiple alignment, which maximizes "information content" within a local window. The idea is that when binding sites are aligned in the window, there will be an "information peak". Another method is called the Gibbs Sampling Strategy, which seeks to find an un-gapped sequence pattern that represents a functional region by calculating a probabilistic model of alignment based on knowledge of the sources of sequence pattern variation and well-established principles of the structure of proteins (Lawrence et al., 1993). The alignment is thus optimized by finding the expectation maximization using the Gibbs sampler (Metropolis, Rosenbluth, Rosenbluth, Teller, & Teller, 1953). One drawback of this method is the assumption that all sequences contribute equally to the conservation calculations. In reality, the distant species that may have lost a particular regulatory function and should not be compared with equal weight to closely related species. Work done by M. Blanchette and M.

Tompa improve on this by incorporating a pre-computed phylogenetic tree and uses a parsimony score to identify motif conservation in a subset of input sequences (Blanchette & Tompa, 2002).

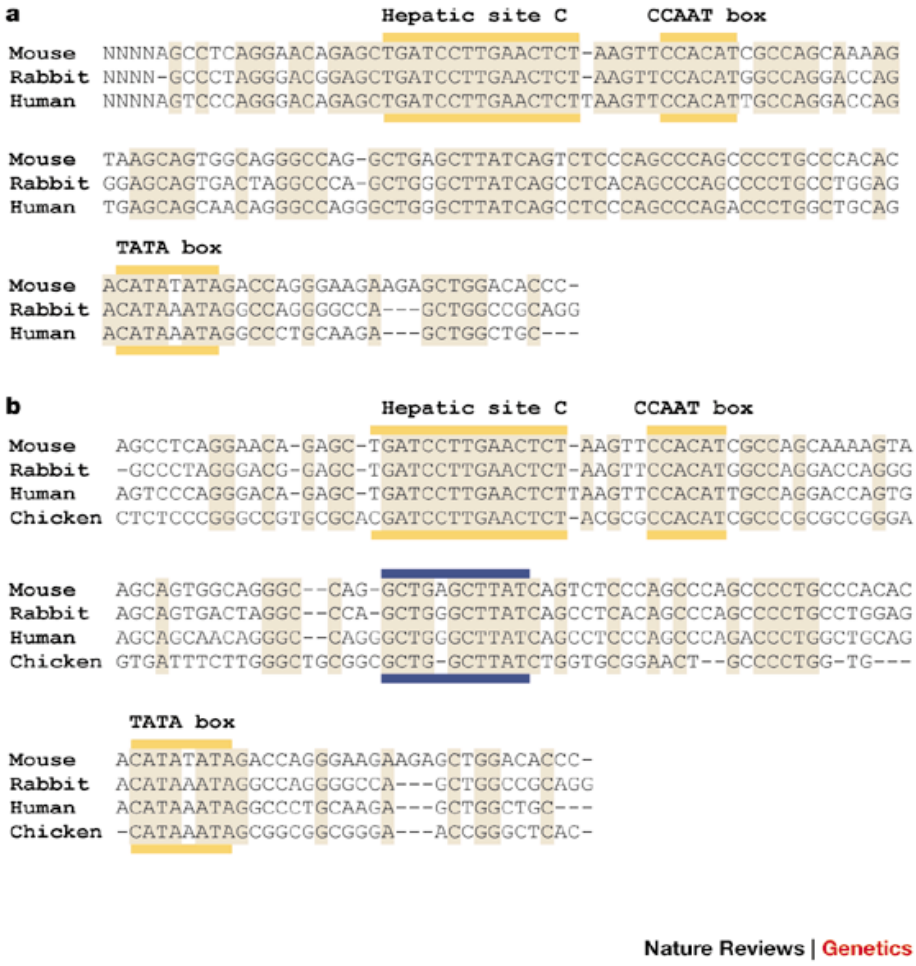



**Figure 9. Illustration of phylogenetic footprinting for the apolipoprotein AI (ApoAI) gene150bp upstream region. (a) Three species sequences are analyzed showing high levels of conservation in a vast majority of the genome (b) Addition of chicken genome decreases the conservation area greatly. Known TFBS are indicated in yellow. Hepatic site C, experimentally shown to be important for ApoAI, is well conserved. Blue region indicate novel conservation region. (Pennacchio & Rubin, 2001).**

#### 1.3.2.1.2 CONTRASIF

Conservation Aided Transcription Factor Binding Site Finder is a tool that is used to perform a genome wide search for TFBS and makes use of evolutionary binding site conservation to filter out false positives. First, a list of orthologous sequences and the TF's motif is required as input into the program. The first

step in CONTRASIF's pipeline is to discover TFBS locations on the genome that matches the TF's motif. This can be done using a PWM or by a Hidden Markov Model Position Weight Matrix (HMM-PWM).

Once TFBS locations are determined, a conservation filter can be applied to reduce false positive predictions. This is done by considering whether a binding site is 1) located in the promoter region, 2) present in the promoter of orthologous genes on other species and 3) above the threshold limit for orthologous gene's peptide identity (which is provided when orthology mappings are imported from Ensembl). If the above three conditions are met, the binding site is kept, otherwise it is removed due to possible false positive.

Currently, running CONTRASIF with conservation filter is limited to a genome-wide promoter search for TFBS on 18 species including human and mouse. Binding sites in enhancers are not considered. This is due to the fact that CONTRASIF does not perform any sequence alignment, thus, it is not possible to determine if detected binding sites on different species align to the same region of the genome; instead, binding sites are only matched if they are detected in the promoter of the same orthologous gene and conservation is analyzed for promoter regions only. Work has been proposed to extend the search to enhancers, but was not yet available at the time this thesis is written.

#### *1.3.2.2 Using Multiple Sequence Alignment of Species*

##### 1.3.2.2.1 rVista

Given a set of aligned orthologous DNA sequences and a TF's PWM from either JASPAR or TRANSFAC, rVista is a program designed to predict the locations of TFBS on the genome and reduce false positive predictions through comparative genomics of multiple aligned species (Loots & Ovcharenko, 2004).

The latest version of rVista (2.0) involves a four-step process. The first step is to identify binding sites on all input sequences using PWMs from the TRANSFAC database (Matys et al., 2003). As this step is often the bottleneck in terms of computation time, tfSearch (Ovcharenko et al., 2005) replaces the previous

MATCH program as it offers a 10-100 fold increase in speed by using a suffix tree based substring search (Loots & Ovcharenko, 2004). Next, local alignment of multiple sequences is performed using Blastz (Schwartz et al., 2003). Pairs of aligned TFBS are then identified and sites that are locally conserved (>80% sequence identity) are selected. Finally, the result is graphically displayed and clusters of TFBS and their conservation profiles are shown in Figure 10.

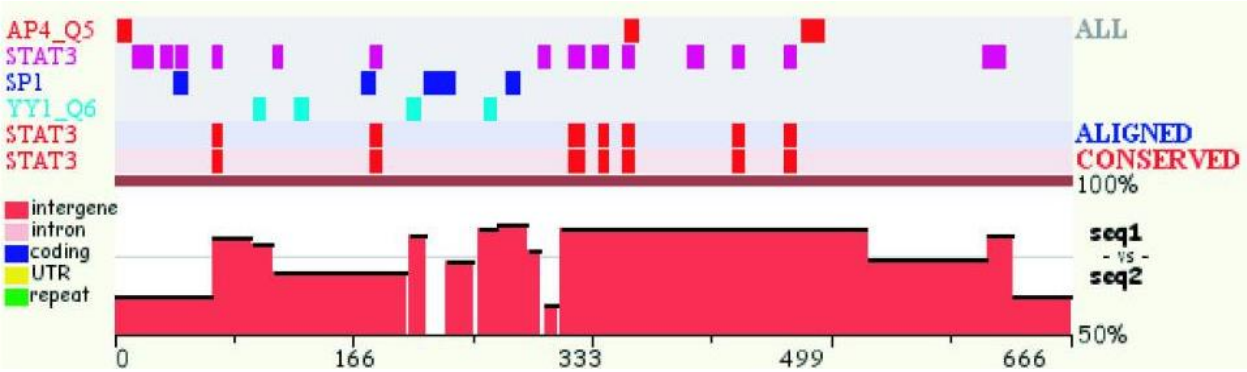



**Figure 10. rVista 2.0 Alignment Visualization (Loots & Ovcharenko, 2004).**

rVista 2.0 can potentially achieve a reduction of up to ~95% in false positive prediction by utilizing a search parameter with high conservation threshold. However, although using a high threshold results in high precision, the method is limited by failing to identify TFBS that are biologically significant to only a few species and not sufficiently conserved in all species sequences. Moreover, the use of alignment algorithms profoundly influences the result of rVista 2.0. It is found that different alignment algorithms align sequences differently and that using more than one algorithm may give a more complete prediction (Bulyk, 2003).

#### 1.3.3.2.2 PReMod

Another approach in the prediction of TFBS is to identify clusters of TFBS on the genome. *cis*-regulatory modules (CRM) are regions of a few hundred base pairs where clusters of TFBS are found. PReMod (Ferretti et al., 2007) is a database of predicted CRM for both human and mouse. CRM predictions are useful in that researchers studying the regulation of a particular gene can use PReMod to look for

nearby regulating CRM. The PReMod database can also be used to identify the binding sites of particular TF where TFBS found within CRM are likely to be true binding sites and clustering of TFBS from different TF may indicate TF-TF interaction.

PReMod follows a two-step algorithm (Blanchette et al., 2006). The first step begins by computationally finding the TFBS of 481 different TF on the human, mouse and rat genome using PWM available on the TRANSFAC database. The use of mouse and rat is to analyze the level of phylogenetic conservation. The genome of these three species are also multiple aligned using MULTIZ. Each identified TFBS is given a log-likelihood ratio score based on a 3$^{rd}$ order Markov model for the background nucleotide frequency that is optimized given the level of surrounding GC content. Each scoring TFBS are then compared with binding sites found on the orthologous position in other species (if no binding site is found, then that species has a TFBS score of zero at that position) and an aligned "hit score" is calculated based on the weighted sum of the scores as shown in Figure 11. High "hit score" is given to sites that have high log-likelihood score and well preserved between the three species. A threshold is used to remove sites with low scores.

The next step is to identify regions that are regulatory modules. This is done by using a sliding window of size 100, 200, 500, 1000 and 2000 base-pairs where for each window, up to 5 top scoring non-overlapping binding site within the window (referred to as "tags") are used to calculate a ModuleScore. The formula for calculating the ModuleScore can be found in (Ferretti et al., 2007). CRM with Module scores > 10 (which corresponds to a p-value < $e^{-10}$) are chosen resulting in 118,402 identified non-overlapping CRM.

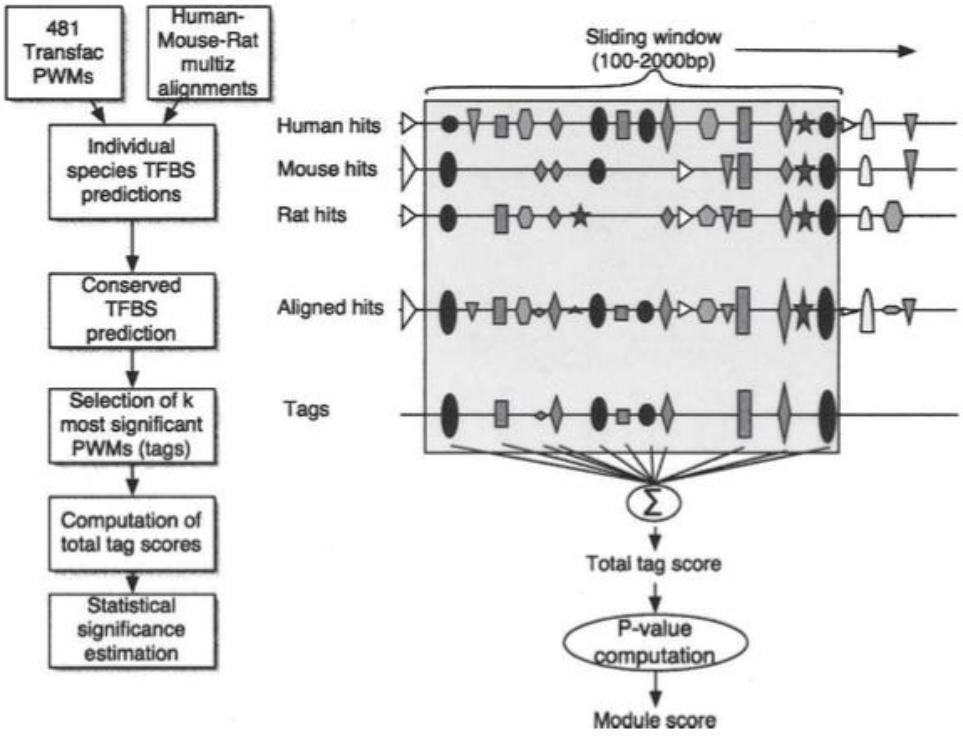



**Figure 11. Example of the PReMod CRM prediction algorithm pipeline (Blanchette et al., 2006).**

### 1.3.3 Machine Learning Methods

In recent years, there have much research done in the use of machine learning (ML) to model and predict the interaction of protein and DNA. Much of the computational work done in the prediction of TFBS lies in pattern recognition and identification of the correlation between biologically relevant binding site and its binding site properties (e.g. location, conservation, etc.). ML algorithms are statistical methods designed to give a computer the ability to learn from experience E such that "its performance on [some task] T, as measured by [some performance metric] P, improves from experience E" (Mitchell, 2006). In our case, the task T is to predict biologically significant TFBS, the performance measure P is the accuracy and experience E is a dataset that includes both known binding and non-binding sites and their properties.

ML algorithms learn information from datasets, which consists of examples and their features. For example, a dataset of fruits could have features {color, taste and size}. An example representing an apple could have a representation such as {color=red, taste=sweet, size=medium}. Learning algorithms are broadly divided into two categories: Supervised and Unsupervised learning. We will briefly discuss the concept of unsupervised learning and delve deeper into supervised learning as our work makes extensive use of algorithms in this category.

#### *1.3.3.1 Unsupervised Learning*

Unsupervised learning is a class of learning problems that aims to find relationships within data (McCrea, 2014). The task is to create meaningful clusters among samples by finding correlation between features. For example, in a police database with features {time, location, crime type}, an unsupervised learning algorithm may detect a cluster of particular crime type being committed at a certain location and time. Figure 12 illustrates the clustering of data of three Iris plant species. Although in Figure 12, it is relatively easy to identify the clusters, clustering becomes much less trivial to human as dimensionality increases. Unsupervised learning algorithms have proven to be useful in understanding the role of TF with regards to the gene regulatory network (Elati & Rouveirol, 2010) among others.

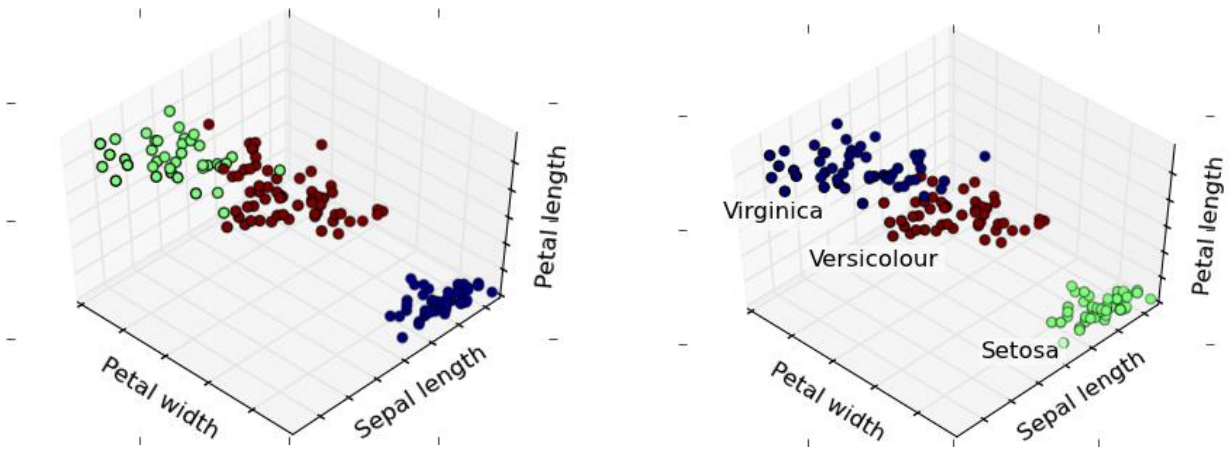



**Figure 12. Visualization of the classification of three types of Iris plants based on their petal width, petal length and sepal length. Computationally determined clusters (left), ground truth (right) (scikit-learn, 2010-2011a).**

#### *1.3.3.2 Supervised Learning*

Unlike the former, supervised learning algorithms learn from a training dataset that is already categorized. The task then is to build a statistical model that predicts the outcome of new data given data already seen. Supervised learning can be subdivided into two more categories: (1) *Regression* predicts an outcome that is continuous. For instance, if 10 hours of practice gives 20% in accuracy and 20 hours of practice gives 40% in accuracy, given 40 hours of practice, a regression algorithm will most likely predict 80% in accuracy if the model generated is linear. (2) *Classification* predicts an outcome that is discrete. For instance, if a 40-hour workweek is considered "acceptable" and 50 hour workweek is considered "unacceptable", a classifier will most likely classify 51-hour workweek as one that is "unacceptable".

Clearly, not all real life examples are linear. In relation to the above example, 80 hours of practice cannot produce an accuracy of 160%; thus, a linear regression algorithm may not be best suited in this case. Similarly, in a classification case, it is not always possible to draw a clear boundary (e.g. 35 hour work week may be considered "unacceptable" as well since less work hour means lower salary). Thus more sophisticated algorithms are used to classify these non-linear cases such as decision tree (Safavian & Landgrebe, 1991), support vector machine (Cortes & Vapnik, 1995) , and Artificial Neural Network (Wang, 2003).

Below we describe two ML algorithms used in our research: logistic regression and support vector machine. In addition, we will discuss cross validation as a method of evaluating an algorithm's performance.

##### 1.3.3.2.1 Logistic Regression

A Logistic regression is a machine learning binary classifier that learns a logistic function where any data, after passing through the function, having a value > 0.5 is labelled as one class while those with a values

< 0.5 are labelled as another class. The logistic function has the form $\sigma(a) = \frac{1}{1+e^{-a}}$ where $a$ is the log odds ratio of the probability of predicting $y$ over the probability of predicting $not\ y$ given the values of the observed features. In logistic regression $a$ is modelled by a linear regression function where each feature is assigned a weight such that, $a = w_0 + w_1x_1 + w_2x_2 + \cdots + w_nx_n$ where $\boldsymbol{w}$ are the weights and $\boldsymbol{x}$ are the features. The linear regression algorithm tries to learn the weights from a training dataset such that each data with $a > 0$ belongs to the same class and $a \leq 0$ belongs to another class (Kim, 2013). New data are classified based on their value after passing through the learnt logistic function.

#### 1.3.3.2.2 Support Vector Machine

A support vector machine (SVM) (Cortes & Vapnik, 1995) is a machine learning classifier that learns a maximum margin hyperplane which separates the training data into distinct regions where each region represents a class. Figure 13 illustrates the classification of two classes (red and blue). The optimal hyperplane is one that maximizes the margin between the hyperplane and the closest data points. In the event that the training data is non-linearly separable, a "kernel trick" can be used to transform the data into higher dimensions such that the data can be separable by a hyperplane. New data are classified based on which region they fall into.

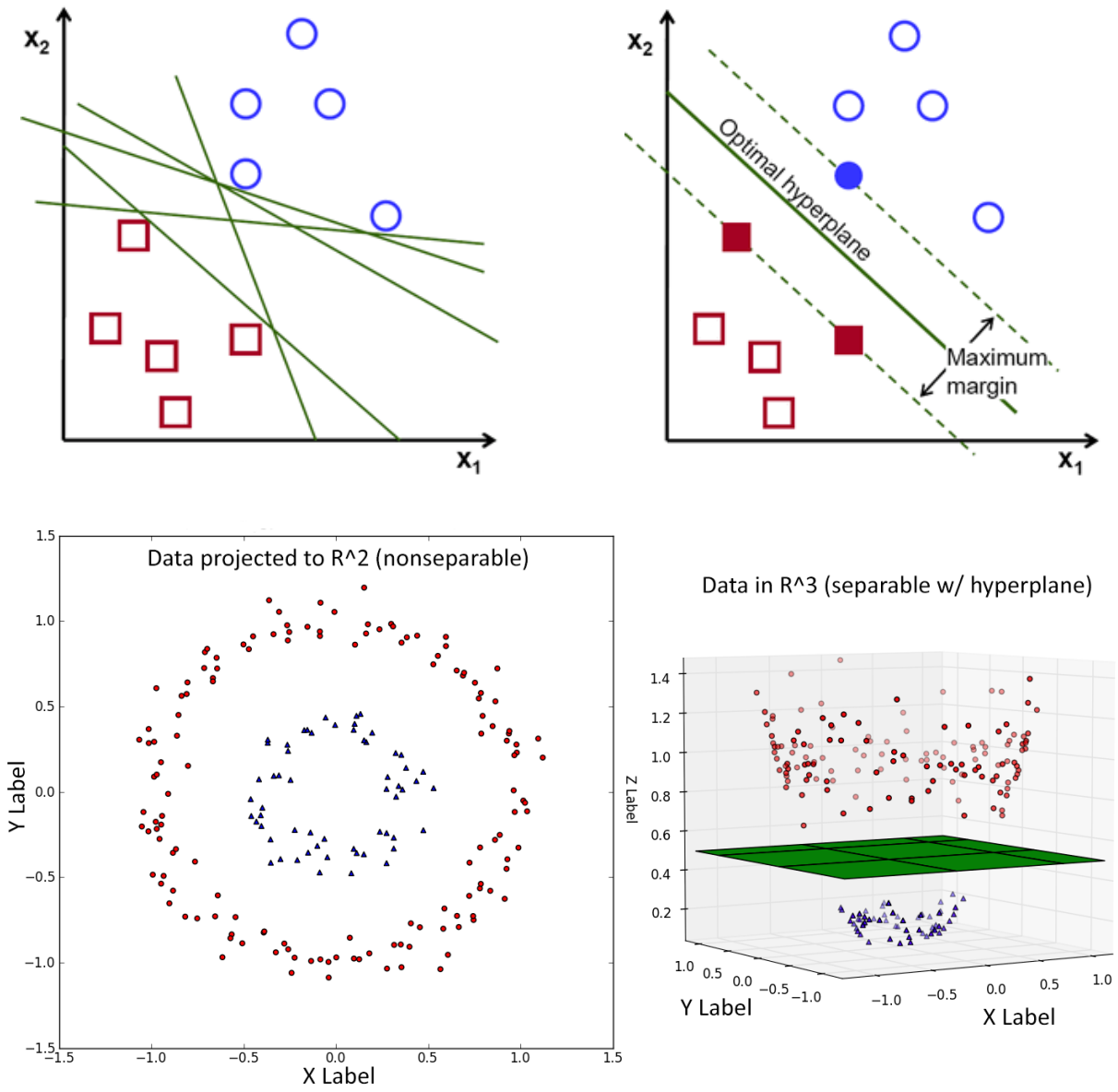



**Figure 13. SVM hyperplane. Many boundary lines can be made (top-left) but the optimal boundary is one that maximizes the margin between the line and the closest data points (top-right). When the data is not linearly separable (bottom-left), a kernel trick can be used to raise the data to a higher dimension such that it is separable by a hyperplane (bottom-right) (Collins, Green, Guttmacher, & Guyer, 2003).**

#### 1.3.3.2.3 Cross Validation

Cross validation is a method of evaluating the performance of a supervised learning algorithm on a dataset without introducing additional test data. This is done by dividing the dataset into two non-overlapping sets such that one is used to learn a ML model while the other is used to test the model's prediction accuracy. Cross validation are typically repeated multiple times using different division of training and test sets. For example, a 10-fold cross validation divides the dataset into 10 even chunks where the evaluation process is repeated 10 times using a different chunk as test set and the rest as training set (Kohavi, 1995).

### *1.3.3.3 Important Considerations*

Recent advances in ML algorithms have spurred great interest in its application. The ability of these algorithms to learn a statistical model without much additional coding makes them cost-effective and popular among data scientists. As more genomic data becomes available, it is increasingly difficult for humans to interpret, and increasingly possible to use ML approaches. However, the requirements for developing a successful ML application go beyond finding the best learning algorithm. In fact, different learning algorithms are often more suited for different situations; for instance, some algorithms perform better when the data is limited while others are better at handling high dimensional data (Caruana & Niculescu-Mizil, 2006). Here we briefly mention a few important considerations when creating a classifier.

P. Domingos wrote that: "Learning = Representation + Evaluation + Optimization" (Domingos, 2012).

*Representation*: A good predictor needs to be able to represent the data. The left most image in Figure 14 shows a predictor that uses a best fit line that is unable to properly represent the data. The predictor in the middle image does a much better job.

*Evaluation*: Some performance measure is required to measure how good a classifier is. This can be an accuracy score, the receiver-operating characteristic (ROC), the maximum likelihood, etc.

*Optimization*: It may not always be possible to get a closed form solution to the optimization problem at hand. Often, algorithms go through iterations refining its own parameters to minimize error and optimization techniques such as gradient descent or greedy search may lead to a local optimum faster than, for example, a constant search. All three aspects are required for an effective classifier.

A good predictor needs to be able to generalize beyond the training data. Setting the learning parameters to greatly minimize error often results in over-fitting. Figure 14 shows an example where the middle image is most likely the proper correlation of the data but would result in some error if tested against the training data itself whereas the image to the right will have a perfect score if tested against the training data but will unlikely predict new data correctly.

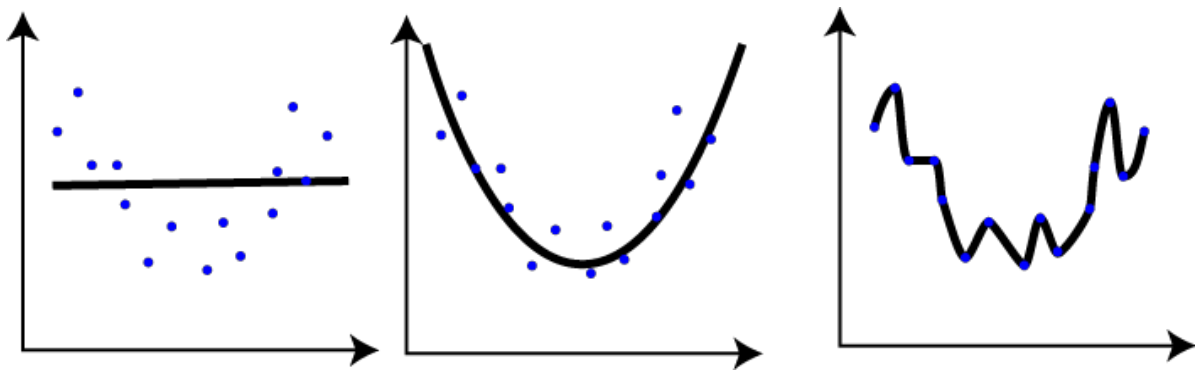


**Figure 14. Three examples of the machine learning fitting the data (black lines are the fit and blue dots are the data). Left: poorly represented and under fitted model. Middle: more accurate model. Right: an over-fitted model (Johnson, 2013).**

Finally, choosing the right features for your dataset is crucial to the predictor's success. Features that correlate well to the outcome make the learning process easier. For example, choosing color as a feature to predict apples or oranges is easier than choosing its size or weight as features. It is common for researchers to spend much more time selecting features, generating the dataset and improving the learner by analyzing the learning result and modifying the data rather than actually running the learning

algorithm itself (Domingos, 2012). More data is often better than better algorithm (Domingos, 2012). As an example, Figure 15 shows how different algorithms can result in equally good classification boundaries given a well-chosen dataset. Finding the features that correlates well with the outcome can be a difficult task, as sometimes multiple features combine to form a good correlation. In our research, we have chosen to use the binding site count within small regions of the genome of 35 different species and their ancestors as features, in the hopes that the learning algorithm will find a positive correlation between binding site clusters in other species and biologically functional binding site in humans.

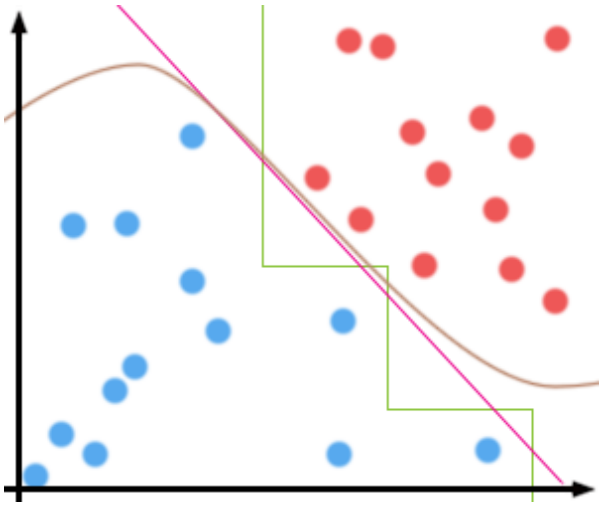


**Figure 15. Learning boundaries of different classifiers on the same dataset. A well-chosen set of samples and features can achieve similar performance given different learning algorithms. The classification boundary of SVM (Brown), Naïve Bayes (Magenta) and Decision Tree (Green) are shown here (Domingos, 2012).**

#### *1.3.3.2 Predicting TFBS by modeling the relationship between Transcription Factors, Transcription Factor's Gene and Transcription Factor Binding Site using ML*

Qian, He and Cai have applied ML, in particular SVM and Nearest Neighbor Algorithm (NNA), in the prediction of TFBS by analyzing the relationships between TF, TFBS, and TF Target (TFT) protein (which are the proteins encoded by the gene in the vicinity of the TFBS) (Ziliang Qian, 2005). The goal is to determine whether a TFBS is functional based on its sequence, the protein properties of the TF and the protein properties of the TFT. As with all ML approaches, the first step is to transform the problem into a

feature/sample dataset matrix. To categorize each TF and TF target proteins, the authors extracted domain information (structural and functional domains) from InterPro giving 8151 unique protein property entries. Each protein is then represented with an 8151 dimensional binary array. The TFBS length is fixed to 25 base pairs. Binding sequence are encoded with zeros and ones where A=00001, C=00010, G=00100, T=01000 and N=10000. Thus a sequence of 25 bp requires 25 x 5 = 125 dimensional array to represent. TRANSFAC is used to extract a positive dataset of TF-TFT-TFBS. Negative data is artificially generated by shuffling the TFBS while keeping TF-TFT fixed. A total of 3430 positive and 7000 negative samples are available. Each sample is a concatenation of [TF][TFT][TFBS] which requires a dimension of 8151 + 8151 + 125 = 16427. Figure 16 shows the performance of using NNA and SVM. We see an average success rate >80% where SVM falls slightly short of NNA.

| Methods | Our Previous Work | | NNA | | SVM (polynomial kernel) | |
|---|---|---|---|---|---|---|
| | Jackknife | 10-fold | Jackknife | 10-fold | Jackknife | 10-fold |
| Success rate on positive dataset | 71.90 % | NA | 84.70 % | 83 % | NA | 76.40 % |
| Success rate on negative dataset | 78.90 % | NA | 89.30 % | 89.10 % | NA | 88.00 % |
| Overall success rate | 76.60 % | NA | 87.90 % | 87 % | NA | 84.10 % |

**Figure 16. TF-TFT-TFBS Performance (Ziliang Qian, 2005). Improvement is seen compared to their previous work which uses only TF-TFBS information (Qian, Cai, & Li, 2006). Jackknife is used here to mean leave-one-out cross validation.**

#### *1.3.3.3 Functional Binding Site Prediction by Integrating Various Genomic Data*

Recent work by Holloway, Kon and DeLisi have shown that SVM can be used to improve the sensitivity and precision of TFBS predictions in 18 eukaryotic genome ranging from human to yeast (Holloway, Kon, & DeLisi, 2005). Given that TF motif matches in the genome often results in false positives, the authors' work examines eight types of genomic data and their usefulness to improve the true positive prediction of TFBS. There is a stronger likelihood that a TFBS is biologically significant if:

1. Binding Site Degeneracy: A greater number of TFBS motif is seen in the promoter of a gene.
2. Conservation: The motif appears in the promoter of orthologous genes in other species.
3. Detection of Clusters: Binding site is within a cluster of binding site from various TFs

4. TF-Target Correlation: The TF has an expression profile that is similar to that of its target gene's expression profile
5. Target-Target Correlation: TF has binding sites in promoter of multiple genes that have similar expression (Yu, Luscombe, Qian, & Gerstein, 2003)
6. GO Annotation: The TF and its target gene have similar functional annotation
7. Phylogenetic Profile: TF's target gene is also detected in orthologous species (Co-evolution of TF and target gene)
8. K-mer distribution: 4-, 5-, and 6-mer distribution patterns are similar to those of regions that contain positive binding sites.

Each genomic data is used as a feature for the SVM classifier both independently and collectively. The result of their experimentation is shown in Figure 17. Combining all features of genomic data and using an SVM classifier gave better prediction than using each genomic data separately.

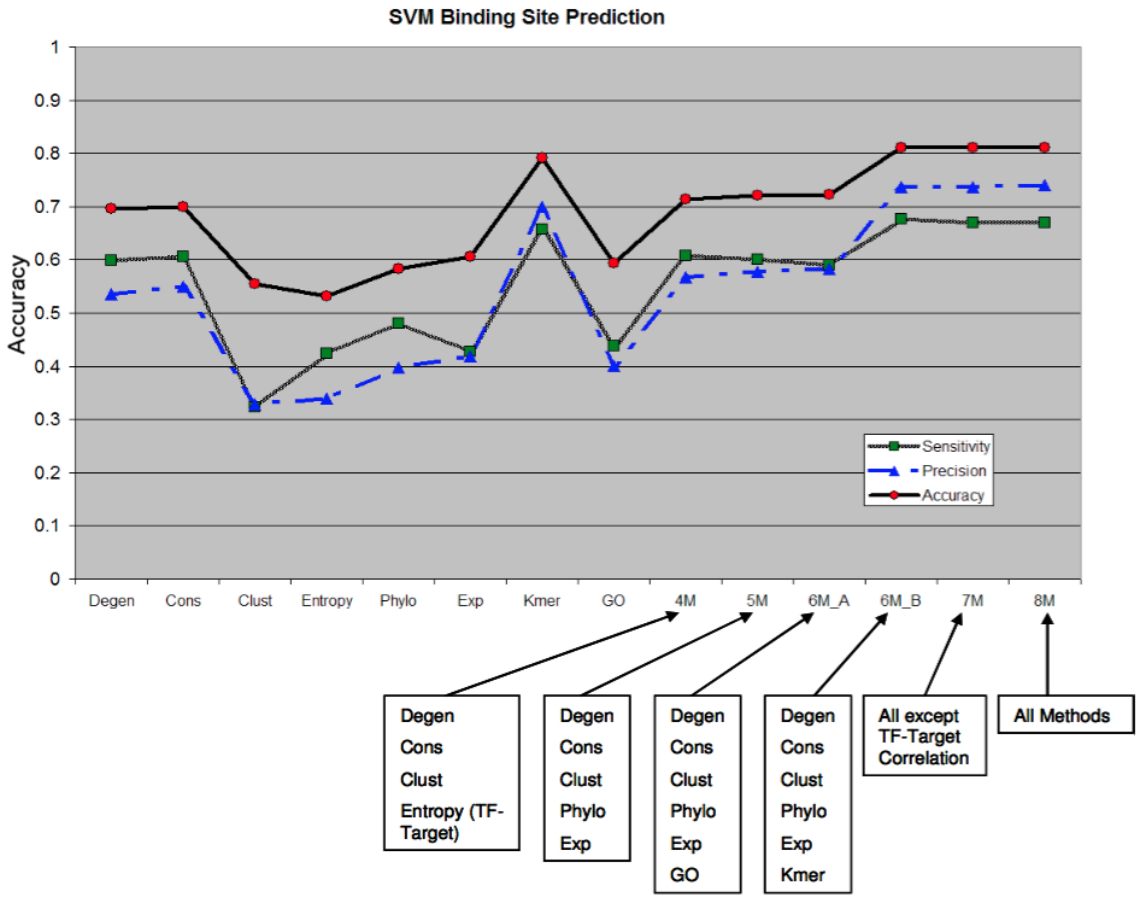



**Figure 17. Accuracy score of an SVM using various genomic data as features (Holloway et al., 2005).**

### 1.3.4 Review of Related Works

Computational prediction of functional TFBS has been an enormous challenge since its inception. Much research has been conducted to improve the accuracy of binding site predictions and some successful methodologies of filtering out false positive predictions include the use of phylogenetic footprinting, PWM refinement and TFBS clustering detection (Bulyk, 2003); all of which have been discussed above. Recent discovery of TF association with nucleosome position, histone marks and hypersensitivity of cleavage by DNaseI has sparked research into methods of TFBS prediction that integrates other genomic data and annotations (Xu et al., 2015) such as the use of genome-wide histone acetylation data (Ramsey et al., 2010), chromatin accessibility data (Pique-Regi et al., 2011) and DNA methylation data (Xu et al., 2015).

In our research, presented in the next chapter, we aim to further improve the computational prediction of TFBS by applying machine learning and using a novel approach of creating a dataset that looks at clustering of multiple TF in the genome of multiple species sequences as well as inferred ancestral genome sequences as a way to detect binding site conservation and to filter out false positive predictions.

# Chapter 2: Predicting Functional TFBS using Comparative Genomics and Machine Learning Methods

It is a well-known fact that genes alone contribute only partially to the encoding of an organism. Portions of the non-coding regions of the genome have been known to act as binding sites for proteins called Transcription Factors. TF binds to specific sequence patterns on the genome to regulate the rate in which nearby genes are transcribed. Much research has been conducted to better understand the role and function of TF with regards to gene regulation. As part of this on-going research, interest has been placed on finding the locations of TFBS. Experimental methods, such as ChIP-Seq (Landt et al., 2012), are powerful techniques used by researchers to study DNA-protein interactions and can be used to accurately locate TFBS on the genome. However, a major drawback of experimental methods is its time and cost. ChIP-Seq experiments require large number of starting material (e.g. DNA oligonucleotides, TFs, antibodies, etc.) and can take from days to weeks to get the result of a single TF (Epigentek, 2015). Recent advancement in sequencing technology has significantly decreased its' cost and made genomic data more readily available. As a result, there is a growing body of research directed in the use of computational methods to find the locations of TFBS. Using information about a TF's motif in the form of a PWM, a computer can quickly scan any specie's sequenced genome for statistical matches. The drawback of computational methods is the rate of false positive predictions. Often, PWM matches are only statistically significant while serving no biological function (Loots & Ovcharenko, 2004). Numerous methods have been proposed to reduce the rate of false positive predictions, such as rVista (Loots & Ovcharenko, 2004) and CONTRASIF (Tokovenko, Golda, Protas, Obolenskaya, & El'skaya, 2009), using phylogenetic footprinting (Ganley & Kobayashi, 2007) and comparative genomics techniques. These methods often sacrifice sensitivity for accuracy. Nevertheless, given the reasonably high accuracy,

the minimal cost and the amount of data that can be mined per unit time, computational methods have proven to be indispensable.

Our research is motivated by the fast-growing field of machine learning and how its ability for pattern recognition can be used in the filtering of false positive TFBS predictions. We use the knowledge that 1) TFBS in promoters are sometimes duplicated due to binding site turnovers (Dermitzakis & Clark, 2002); thus, the more instances of a TF's bind sites that are located near each other, the more likely the binding site is functional. 2) Different TF often work together and the detection of binding site clusters is a good indication of biological significance. 3) Functional binding sites tend to experience positive evolutionary pressure (Holloway et al., 2005). If a TFBS is functional, it will most likely show up in similar species as well. We then take a ML approach to model correlation between TFBS locations and their biological significance. Previous work done by Blanchette et al. (Blanchette, 2012) (as discussed above) uses a window size of 200 to scan across chromosome 1 in the human genome and the number of predicted TFBS within the window is counted. The same window is also scanned across multiple other specie sequences and "pre-computed" ancestral sequences. A log-likelihood score is then given based on how probable the number of predicted TFBS count is preserved between different species given their divergence. The score with a threshold is used to determine if the TFBS is a false positive.

We build upon their research by taking into account all PWM-matching TFBS in all human chromosomes. We also consider multiple TF simultaneously and apply ML to determine if there are any interactions between them. The following sections will outline the data, tools and algorithms used to make the analysis followed by a discussion of our findings.

## 2.1 Data Sources and Method

### 2.1.1 Overview

Three sources of data were used in the development of our predictor: Regions of DNA-TF interaction in human produced by ENCODE ChIP-Seq experiments (ENCODE Data), genome positions of TF motif matches in multiple species and their ancestors (PWM Data) and regions of CpG islands (CpG Data).

These data are then used to create a ML classifier that will tell us whether a computationally identified TFBS on the human genome has any real biological function. Figure 18 shows a flowchart of our methodology. The first step is to parse the data files into computer readable data structures. Then the ENCODE and PWM data are processed to create a ML dataset that counts the number of binding sites around each of the PWM inferred binding sites found in human. When necessary, the CpG data is used to split the original dataset into a smaller dataset that includes samples found in CpG islands only or vice versa. A machine learning algorithm then uses either the full dataset or the split dataset to build a functional TFBS predictor.

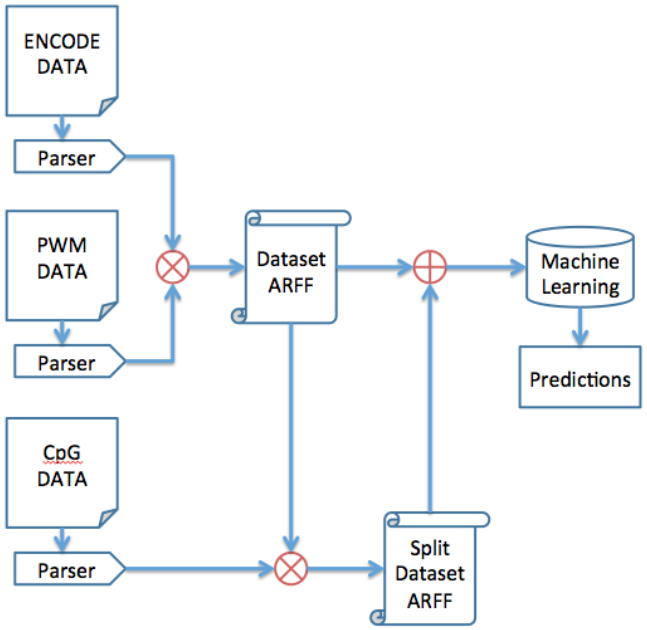



**Figure 18. Flowchart of our TFBS predictor. Red X indicate a logical "AND". Red + indicate a logical "OR".**

This section will start by introducing the method of reconstructing ancestral sequences from extant species as we consider the binding sites found in these genomes in our predictor as well. We then present the ENCODE, PWM and CpG data and the parsing of these data, followed by a detailed description of how our dataset is generated. We end this section by discussing the tools used to perform machine learning and the metrics used to evaluate its performance.

### 2.1.2 Reconstruction of Ancestral Genome

The data we use contains not only the TFBS locations on 35 mammalian species sequences, but also on 34 reconstructed ancestral genome sequences (Figure 19). While the mammal genomes are sequenced through classical methods, ancestral genomes for each of the internal node of the phylogenetic tree are reconstructed computationally (Miller et al., 2007). Work by Blanchette et al. has shown that given a phylogenetic association between sufficiently many well-chosen extant mammalian genomes, it is possible to achieve a reconstruction with up to 99% base-by-base accuracy (Blanchette, Green, Miller, & Haussler, 2004), as discussed below.

The reconstruction aims to infer genomic sequence evolution in terms of insertion, deletion and substitution. The challenge is multi-layered and begins with the construction of an accurate whole genome multiple sequence alignment of the chosen species. Much work has been done in this respect and among them is the Threaded Blockset Aligner (TBA) (Blanchette, Kent, et al., 2004). This method differs from the typical approach of using a reference genome, which cannot identify conservation of regions that are not present in the reference genome; instead, it uses a “block” approach where alignment is performed based on all sequences in the block. TBA is developed using a collection of independently executing software including BLASTZ for pairwise alignment, RepeatMasker (Tempel, 2012) and MULTIZ (Blanchette, Kent, et al., 2004) for alignment between three or more sequences.

The sequence alignment, along with a phylogenetic tree whose branch lengths are inferred using the HKY model (Hasegawa, Kishino, & Yano, 1985) and PHYML program (Guindon & Gascuel, 2003), is then used to identify which alignment position belongs to an ancestor and which are insertions and deletions. For each position that is predicted to belong to an ancestor, context-dependent maximum likelihood estimation is used to determine identity of the nucleotide that position (Blanchette, Kent, et al., 2004).

It should be noted that not all ancestral sequences can be reconstructed with the same accuracy. An ancestor that has, in a short time, diversified into a large number of independent descendant species is better suited for reconstruction. In fact, instantaneous radiation of an ancestor into numerous species can in theory achieve a reconstruction accuracy that approaches 100% (Blanchette, Green, et al., 2004). The boreoeutherian ancestor is close to such an example and simulation of the ancestral sequence reconstruction (using pre-computed ancestral sequences) reveals a reconstruction accuracy of 99%. The rapid radiation of species from the boreoeutherian ancestor makes reconstruction of early eutherian ancestors in the eutherian mammal phylum highly accurate.

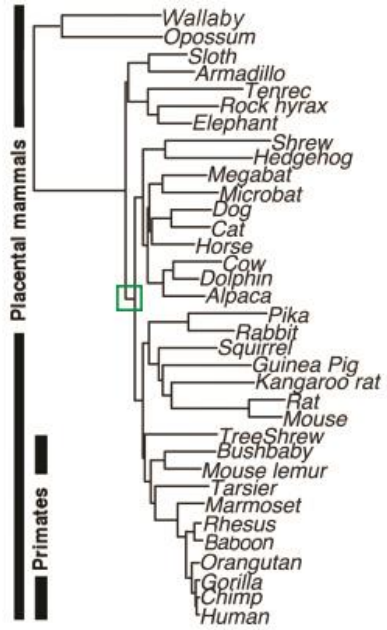



**Figure 19. An updated version of the phylogenetic tree by (Miller et al., 2007; Sadri, Diallo, & Blanchette, 2011) showing 35 species and 34 ancestral nodes. The Boreoeutherian ancestor is highlighted with a green box.**

### 2.1.3 TF-DNA Interaction Regions in Human

#### *2.1.3.1 Data*

It is estimated that only about 1.5% of our DNA code for genes. The ENCODE project, conceived in 2009, is an international effort to try and identify the function of the rest of the human non-coding genomic regions. Among their experimentations is the mapping of binding locations of 119 DNA-binding proteins using ChIP-Seq where 87 of them were TF (Consortium, 2012). Of the 87 TF-DNA binding data, 46 of them had both ChIP-Seq and PWM data available to us. After further selection (details in Appendix-1), a subset of 38 was used in our analysis. Data are separated by TF. Each file contains a list of ranges that marks the location in the human genome where a TF-DNA interaction is observed. Since ChIP-Seq cuts DNA into short variable length strands, the exact position within this region where binding occurs is unknown; however, there is a high probability that a binding did occur somewhere within this region. It is possible for different cell lines in human to exhibit different TF-DNA interaction patterns. In the case of the ENCODE data, it is an accumulation of all detected binding sites in approx. 100 cell lines in human.

#### *2.1.3.2 Data Processing*

Parsing of the ENCODE data files, which are the locations of our "positive" (or functional) binding site regions, is done by converting the files into a Python *dictionary* that maps a chromosome number to a list of (start, end) Python *tuples*. Each tuple marks the starting and ending index of a TF-DNA interaction region. Each dictionary item's list is sorted based on starting index. The result is the following data structure: *dictionary key(chromosome number) -> list**(**region1{start, end}, region2{start, end}, …**)***

### 2.1.4 PWM Scanning for TFBS

For each TF, their motif (represented by a PWM, some chosen from the TRANSFAC database and others from the JASPAR database) is scanned across the genome of multiple species (Figure 20) to find positions of high likelihood matching. The tool used to scan the genome for PWM matches is an in-

house program that uses a classical log-likelihood approach (discussed in 1.3.1). The resulting data is a list of positions separated by TF. Each position is associated with one of 69 species/ancestral sequences, thus encompassing the predicted TFBS for all specie

| Species | UCSC Version |
|---|---|
| **Human** | HG19 |
| **Chimp** | PANTRO2 |
| **Gorilla** | GORGOR1 |
| **Orangutan** | PONABE2 |
| **Rhesus** | RHEMAC2 |
| **Baboon** | PAPHAM1 |
| **Marmoset** | CALJAC1 |
| **Tarsier** | TARSYR1 |
| **Mouse Lemur** | MICMUR1 |
| **Bushbaby** | OTOGAR1 |
| **Tree Shrew** | TUPBEL1 |
| **Mouse** | MM9 |
| **Rat** | RN4 |
| **Kangaroo Rat** | DIPORD1 |
| **Guinea Pig** | CAVPOR3 |
| **Squirrel** | SPETRI1 |
| **Rabbit** | ORYCUN2 |
| **Pika** | OCHPRI2 |
| **Alpaca** | VICPAC1 |
| **Dolphin** | TURTRU1 |
| **Cow** | BOSTAU4 |
| **Horse** | EQUCAB2 |
| **Cat** | FELCAT3 |
| **Dog** | CANFAM2 |
| **Microbat** | MYOLUC1 |
| **Megabat** | PTEVAM1 |
| **Hedgehog** | ERIEUR1 |
| **Shrew** | SORARA1 |
| **Elephant** | LOXAFR3 |
| **Rock Hyrax** | PROCAP1 |
| **Tenrec** | ECHTEL1 |
| **Armadillo** | DASNOV2 |
| **Sloth** | CHOHOF1 |
| **Opossum** | MONDOM5 |
| **Wallaby** | MACEUG1 |

**Figure 20. Names and UCSC Versions of the 35 species used in our analysis.**

In addition, all species/ancestral sequences are aligned to the human sequence. This means that the alignment positions and human positions are equivalent. To illustrate this, Figure 21 shows a sample alignment between human, mouse and rat. We see that in order to align to the human sequence, the 4 additional nucleotides that exist in mouse and rat, but not in human, as represented by the gaps, are truncated. It is possible that there will be positions in human that doesn't exist in other species due to differences in genome size. Similarly, there could also be TFBS that exist in regions of the genomes of other species that does not align to human. In our work, we are not concerned with regions that do not have a correspondence in the human genome as our focus primarily lies on the prediction of functional TFBS in human.

Due to the difference in length and specificity between different TF motifs, the number of inferred binding sites in human vary greatly between TF, ranging from a few thousand for TFs such as NRF-1 to millions for TFs such as YY1. All binding sites found using PWM scanning will henceforth be referred to as “inferred binding sites”.

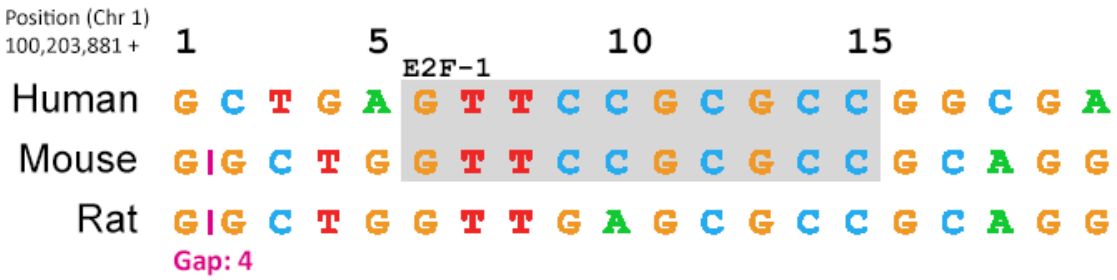



**Figure 21. Example of multiple sequence alignment between human, mouse and rat. Alignment shows the location of a possible E2F-1 binding site in human chromosome 1 (grey box). The purple vertical dashes indicate there are gaps in the alignment where there are additional nucleotides in mouse and rat that does not exist in human. However, since they do not align to human, they are truncated such that the alignment positions match the human positions.**

Parsing of the PWM-based binding site files, which contains locations of motif matches in 69 species/ancestral sequences, is done by converting the files into a Python *dictionary* that maps a specie number to a list of motif matching locations in that specie’s genome. Each dictionary item’s list is sorted by the value of the index position. The result if the following data structure:

*dictionary key(specie number) -> list**(**index1, index2, …**).***

### 2.1.5 CpG Island Locations in Human

CpG islands are short interspersed DNA sequences that are predominantly non-methylated and are commonly sites of transcription initiation (Deaton & Bird, 2011). We obtained the human genome CpG island annotation from the UCSC genome browser database, where it was computed based on large-scale epigenome predictions (Bock, Walter, Paulsen, & Lengauer, 2007). We use this data to see whether TFBS prediction accuracy has a dependence on whether or not the binding site is within a CpG island.

Parsing of this file is done similar to parsing both the ENCODE and PWM scanned files where the CpG island files are converted to a Python *dictionary* that maps chromosome number to a list of ranges identifying regions of CpG islands. The lists are sorted based on range's starting index. The result is the following data structure:

*dictionary key(chromosome number) -> list**(**range1{start, end}, range2{start, end}, …**)***

### 2.1.6 Dataset Generation

In any ML application, the careful creation of the dataset is often the most important step to its success. A poor choice of features and population to sample can cause the outcome to be biased and non-representative. We want to generate a ML dataset that can model the clustering of multiple TFBS as well as its evolutionary conservation. Since we are only interested in the prediction of functional TFBS in humans, for every TF, the list of every inferred binding site in human is the sample space. For each sample, we label the binding site as functional (binding occurs *in vivo*) if it is within a region experimentally determined by ENCODE to show TF-DNA interaction and non-functional (binding does not occur *in vivo*) otherwise. We then take a fixed window size of width *w* nucleotide bases to the left and right of each inferred binding site on the human genome and count the number of predicted sites found in human and in each other species/ancestor within the same window as illustrated by Figure 22. The use of this window allows us to observe any clustering of TFBS nearby and detect if the same binding site location is conserved in other species. It also takes into account TFBS turnover as it is observed that new binding sites tend to emerge near existing functional binding sites (Dermitzakis & Clark, 2002). Next, for each TF we analyze, we take the same window and extend it to the 37 other TF and count the number of inferred binding sites in all species as well. This results is a feature size of 69 species/ancestors x 38 TF = 2622 total. This dataset is then fed into a learning algorithm to learn the difference in binding site concentration pattern between functional and non-functional TFBS.

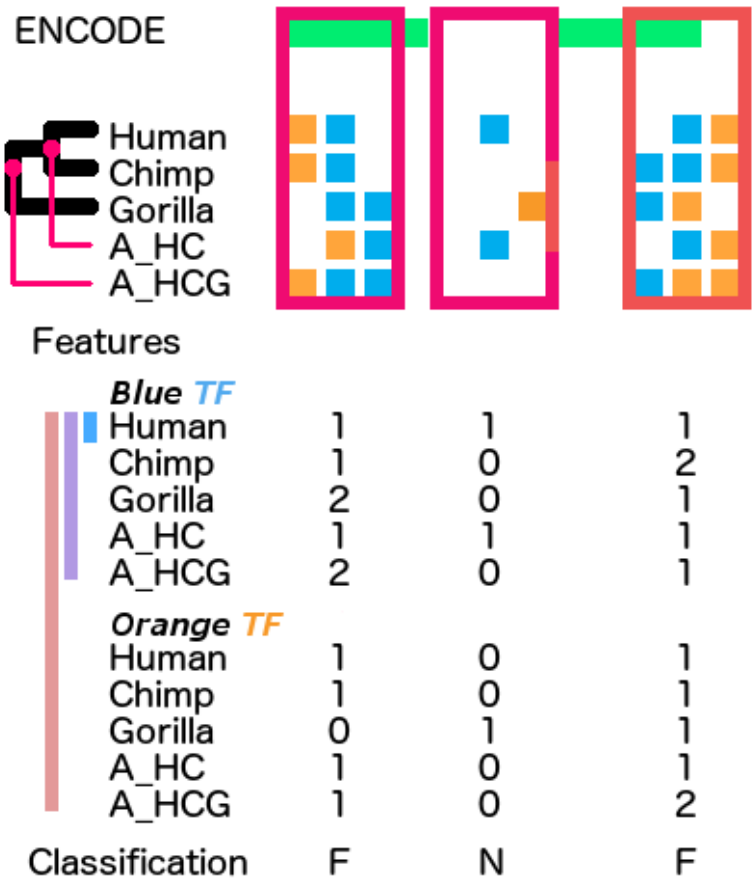



**Figure 22. Visualization of our dataset generationprocedure on a toy example with two TFs and 3 species and 2 ancestors. "A_" are ancestor sequences. "A_HC" represents ancestor of human and chimp and "A_HCG" represents ancestor of human, chimp and gorilla. Assume orthologous sequences are aligned and horizontally stacked on top of each other. Blue and orange squares represent the PWM matches for two different TF. In this example, we are interested in building a data set for the blue TF; thus, a window (red) is draw around every match for the blue TF found in human. The green bar on top indicate regions on the human genome where ENCODE ChIP-Seq experiment shows evidence of TF-DNA interaction for the blue TF. Every candidate binding site that is within the green region is classified as "Functional"; "Non-Functional" otherwise. Each red window is a sample and for each sample, the number of candidate binding sites for all TF are counted. The actual dataset contains 35 species, 34 ancestors and 38 TF. Illustration of the amount of data encompassed by our three datasets (discussed in section 2.2) is shown by the colored lines to the left of the features: SS-ST (Blue), MS-ST (Purple) and MS- MT (Red).**

### 2.1.7 Algorithm for Assembling the Data Set

Due to the large number of ENCODE genomic intervals (up to ~80,000 per TF) and inferred binding sites (up to ~2 million per TF per species), assembling the data set is challenging because it is impractical to

linearly search through every species to count binding sites that are within a given window. Given that multiple binding sites could overlap with the window, all binding sites in all species must be searched to ensure none are missed. This is then repeated for every window. However, by realizing that since the binding sites and window ranges are known ahead of time we could avoid random searches by sorting them such that we search in increasing order. We propose an algorithm that will generates the dataset in linear, ***O****(n + m),* time where *n* is the total number of binding sites in all species and *m* is the number of ENCODE regions. The following algorithm assumes all required data are parsed and sorted by position:

*Preliminary:*

- **E**[*tf*] ENCODE chip-seq data. Given a TF *'tf'*, returns a list of ranges where TF-DNA interaction occurred for that TF in the form of a tuple {start, end} indicating the starting and ending index of that range. Sorted by start value in the tuple.
- **P**[*tf*][s] Positions of TF motif matches on the genome. Given a TF '*tf'* and a species *'s'*, returns the list TFBS positions. The list of positions is sorted for each species.
- e_index Integer keeping track of the index we are currently looking at for the ENCODE regions. Initialized to zero.
- p_index[*tf*][s] Integer array keeping track of the index we are currently looking at for species '*s'* in transcription factor data '*tf'*. Initialized to zero.
- b_count[*tf*][s] Integer array keeping track of the TFBS count for each species 's' in transcription factor data '*tf*'. Initialized to zero.
- dataset[][] Final output.

```
FOR TF_CURRENT IN List of all 38 Transcription Factors:
    // 1. Initialize variables and an empty dataset for TF_CURRENT
    e_index = 0
    dataset = [][]

    FOR POS IN All positions found in P[TF][HUMAN]:

        // 2. Determine if position is functional or non-functional
        WHILE POS > E[TF_CURRENT][e_index].end:
            e_index++

        IF POS > E[TF_CURRENT][e_index].start:
            sample = [1]  // New integer array with one entry containing the value 1
        ELSE:
            sample = [0]  // New integer array with one entry containing the value 1

        // 3. Define a window size
        windowMin = POS - w
        windowMax = POS + w

        // 4. Count binding sites for all species in all TF
        FOR TF IN List of all 38 Transcription Factors:
            FOR SPC IN List of all 69 species and ancestors:

                Set all entries in b_count to zero
                spc_index = p_index[TF][SPC]

                // 5. Rewind index
                WHILE P[TF][SPC][spc_index] > windowMin:
                    spc_index - -

                // 6. Move index forward
                spc_index++
                WHILE P[TF][SPC][spc_index] < windowMax:
                    b_count[TF][SPC]++
                    spc_index++

                p_index[TF][SPC] = spc_index

        // 7. Add sample to dataset
        Horizontal_Concatenate(sample, b_count)
        Vertical_Concatenate(dataset, sample)

    Export(dataset)
```

Despite the nested loops, most inferred TFBS positions on each species for each TF are checked only once. Binding sites belonging to multiple windows will cause a rewind equal to the number of windows they belong to. The idea is to recognize that when positions and ranges are sorted, if the current position is larger than the upper boundary of an ENCODE range or window range, all future positions will be larger than these ranges and we can discard them. Since it is possible for multiple positions to fall into the same window range, we need to rewind slightly (step 4) to make sure the positions we have seen are not within the new window range. However, this only happens when binding sites are close and windows are overlapping, which is not frequent. The other time consuming operation is to sort all the data, which in our case turns out to be negligible. This is because the data themselves are already either sorted or mostly sorted. The Timsort algorithm (Peters, 2002) used to sort the positions and ranges performs exceptionally well for our situation. Timsort detects chunks for sorted region within the unsorted list and sort the chunks. If the list is already sorted, Timsort does no additional work.

### 2.1.8 Development of the Predictor

To develop the predictor for our dataset, we used two ML libraries: Scikit-Learn and Weka.

#### *2.1.8.1 Scikit-Learn*

Scikit-Learn is an open source ML library for the Python programming language (Pedregosa et al., 2011). It includes a plethora of learning algorithms including the one of particular interest to us, SVM. Although Python is an interpreted language, which is slower in execution compared to compiled languages (e.g. C++), many of the algorithm implementations are in C and uses a bytecode interpreter called Cython (Behnel et al., 2011) to interface with C/C++ code to improve performance. For example, Scikit-Learn uses LIBSVM (Chang & Lin, 2011), an open source C++ library for SVM developed at National Taiwan University, as its underlying SVM implementation and is programmed in C/C++. We use Scikit-Learn for training classifiers, cross validation and performance evaluation.

#### *2.1.8.2 Weka*

Weka is a Java-based ML software developed at the University of Waikato, New Zealand. The software comes with a graphical user interface as well as a full set of APIs. Weka reads a specially formatted file format called Attribute-Relation File Format (ARFF) as its standard dataset input (Hall et al., 2009). ARFF is created specifically for Weka but has since increased in popularity and parser for ARFF is available in many programming languages, including python. Many other ML libraries have also adopted the use of ARFF as their dataset input format. Although ARFF is not directly supported by Scikit-Learn, in the development of our classifier, we provide implementation to use ARFF as our standard input and output dataset format. Due to Weka's well-documented libraries and the ability to analyze the predictor model generated by the learning algorithm, we use Weka to perform Simple Logistic Regression as a way to analyze the impact different species and TF have on the prediction outcome by looking at the assigned weights.

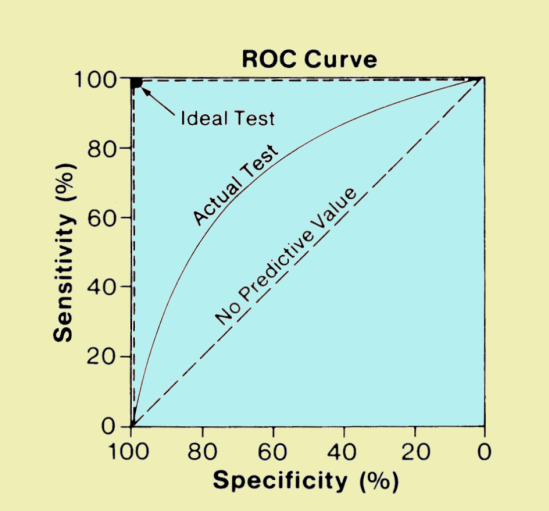
#### *2.1.8.3 Dataset Balancing*

After the initial dataset generation, we observed that ~95% of all inferred binding sites are negative examples, i.e. they do not fall within any ENCODE regions. As a result, simply learning to predict all examples as negative, our classifier would obtain an accuracy above 95%. This situation makes learning better classifiers difficult. To counteract this issue, we randomly remove negative samples until there is a balance between the number of positive and negative samples. It is possible that by omitting some negative samples, its patterns will not be learnt; but if enough distinction exists between positive and negative samples, this will only result in small fluctuations in the outcome of our classifier. All the results we report are the average over at least 3 different negative. The number of examples in the balanced data set ranges from ~500 for PBX3 to 45,000 for AP-2 Gamma, with an average of ~10,000.

#### *2.1.8.4 Evaluation Metric and Area under the ROC Curve (AUC)*

To evaluate the performance of our classifiers, we use the Area Under a Receiver Operating Characteristic (ROC) curve (AUC) (Sprawls, 1995). The ROC curve is a graph of True Positive Rate (TPR aka Sensitivity) vs. False Positive Rate (FPR, also 1-Specificity) that is used to show how well a binary classifier performs at different values of the decision threshold. The sensitivity measures the percentage of all positive samples that are predicted correctly while the specificity measures the percentage of all negative samples that are predicted correctly. The FPR then measure the inverse of specificity, which is the percentage of all true negative samples that are predicted incorrectly.

By varying the decision threshold of classifiers, we typically either increase the number of positive predictions and decrease the number of negative predictions or vice versa. When the number of positive predictions are increased, the TPR and FPR generally increases giving us a point towards the upper right corner of the graph; similarly, when there are fewer positive predictions, TPR and FRP are lower, giving us a point in the bottom left region. The typical plot that result from this change in threshold is seen in Figure 23 as “Actual Test”. The area under ROC is calculated by taking the integral over the x-axis. A perfect predictor will have 100% TPR and 0% FPR giving us a point on the top left. Integrating this curve will result in a value of 1. An area under ROC value of 0.5 indicate a predictor that is as good as randomly guessing the label, while a value below 0.5 is a predictor that tend to guess wrong. Nevertheless, the inverse of a predictor that tends to guess wrong is one that tends to guess right!



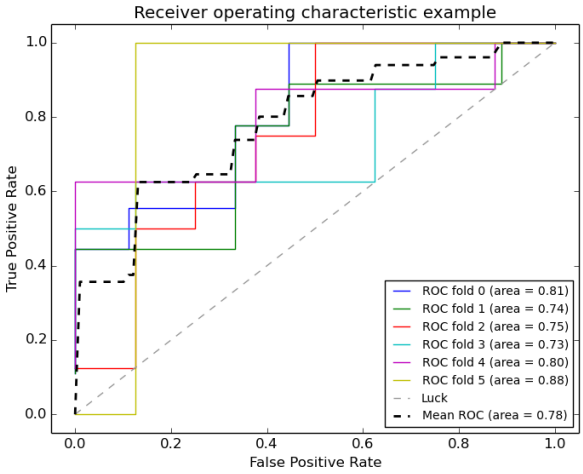




**Figure 23. Illustration of a ROC curve; note the inverse in value on the x-axis (left). An example of an ROC curve used to measure the cross-validation performance of binary classifiers in scikit-learn (right) (scikit-learn, 2010-2011c; Sprawls, 1995).**

### *2.1.8.5 Area under the ROC Curve for Cross Validation*

Since our goal is not to determine how well a classifier performs with varying threshold, but how well it performs based on cross-validation, the area under the ROC curve is calculated in a slightly different way by the scikit-learn library (scikit-learn, 2010-2011c). An example of a ROC curve computed using a 6-fold cross-validation is shown in Figure 23. For each dataset, the samples are first evenly divided into 6 separate chunks. We put one chunk aside and train a classifier using the remaining 5 chunks. A prediction, along with a confidence level of this prediction, is made on each of the samples in the unused chunk using the trained classifier. Since we know the true output of each of the samples in the unused chunk, we can compare the prediction with the true output. By adjusting the threshold of the confidence level, we get a set of different TPR/FPR values. The TPR/FPR values are then used to compute a ROC curve. This is repeated 6 times, removing a different chunk of the dataset each time. The result is 6 different ROC curves. The AUC for each ROC curve is calculated and the mean AUC is determined by taking their average.

## 2.2 Results and Discussion

The family of predictors we have developed aim at determining whether a given position *p* of the human genome that is match for the PWM of a given transcription factor *T* is actually going to be bound by that TF in at least one human cell type. In other words, its goal is to separate true from false positives among TFBS predictions obtained by PWM scanning. We consider a family of predictors that uses increasingly rich sets of features:

1) The **baseline predictor** assumes that all human sites with a match to the PWM for *T* will indeed be bound by that *T*, i.e. it predicteds as positive all examples presented to it.
2) The **single-species single-TF** (SS-ST) predictor attempts to identify true binding sites for T in the human genome by considering the number of PWM matches for T in a window of 450bp centered around the PWM match of interest (the choice of the window size of 450bp is justified in Section 2.2.2). Since many transcription factors tend to have multiple binding sites in the same promoter or enhancer (Dermitzakis & Clark, 2002), there is hope that this classifier may be able to perform better than the baseline predictor. Dimensionality of feature space: 1.
3) The **multi-species single-TF (MS-ST) predictor** complements PWM match counts in human with PWM match counts in regions that are orthologous to the 450-bp human region in a set of 34 other mammalian species and their 34 computationally predicted ancestors. Since functional binding sites tend to be conserved during evolution (although they are often subject to turnover), this additional evolutionary information may allow this predictor to outperform the single-species single-TF predictor. Dimensionality of feature space: 69.
4) The **multi-species multi-TF (MS-MT) predictor** considers an even larger set of features, by complementing the information available to the multi-species single-TF predictor with the analogous information for all 38 transcription factors other than T. Since functional binding sites for transcription factor T are often surrounded by predicted binding sites for other TFs, one may

hope that this predictor may be more accurate than the MS-ST predictor. Dimensionality of feature space: 38 TF x 69 Species = 2622.

Each predictor is trained on a balanced training set where negative examples were downsampled to match the number of positive examples (downsampling was repeated 3 times; all numbers reported are the average over these 3 repetitions). In all cases, predictors were evaluated using 10-fold cross-validation.

### 2.2.1 Choice of Classifier

To decide on a classifier that would best represent our data, we tested 10 different ML algorithms found in the scikit-learn library and compared their prediction accuracy against our MS-ST dataset. Each algorithm's default parameters were used and these default values can be found on the scikit-learn website (Pedregosa et al., 2011). The result is listed in Table 1. We see that SVM has the highest AUC accuracy score for 73% of all TF, followed by logistic regression having the highest score for 21% of all TF. A closer examination shows that SVM is never too far behind in performance compared to other algorithms having a higher accuracy. Thus, we choose these SVM as our algorithm of choice. For SVM, we use the Radial Basis Function kernel (scikit-learn, 2010-2011b) with the "cost of classification – C" equals to 1.0 and a gamma value of $\frac{1}{number\ of\ features}$.

| | Logistic Regression | K Neighbors | Nearest Centroid | LDA | SVM | Random Forest | Decision Tree | Multinomial Naive Bayes | Gaussian Naive Bayes | Bernoulli Naive Bayes |
|---|---|---|---|---|---|---|---|---|---|---|
| NF-YA | 0.893 | 0.855 | 0.855 | 0.875 | 0.897 | 0.880 | 0.747 | 0.726 | 0.736 | 0.658 |
| Nrf1 | 0.871 | 0.834 | 0.834 | 0.868 | 0.878 | 0.869 | 0.747 | 0.772 | 0.811 | 0.849 |
| ELK4 | 0.794 | 0.788 | 0.788 | 0.761 | 0.839 | 0.827 | 0.740 | 0.762 | 0.781 | 0.792 |
| SP1 | 0.818 | 0.765 | 0.765 | 0.815 | 0.803 | 0.784 | 0.619 | 0.736 | 0.727 | 0.769 |
| E2F1 | 0.804 | 0.732 | 0.732 | 0.793 | 0.818 | 0.781 | 0.671 | 0.716 | 0.726 | 0.793 |
| Egr-1 | 0.803 | 0.722 | 0.722 | 0.798 | 0.796 | 0.764 | 0.607 | 0.695 | 0.741 | 0.758 |
| ATF3 | 0.752 | 0.710 | 0.710 | 0.671 | 0.783 | 0.792 | 0.693 | 0.727 | 0.704 | 0.746 |
| AP-2alpha | 0.779 | 0.708 | 0.708 | 0.778 | 0.750 | 0.726 | 0.583 | 0.646 | 0.708 | 0.719 |
| GABP | 0.750 | 0.685 | 0.685 | 0.733 | 0.778 | 0.737 | 0.634 | 0.677 | 0.720 | 0.747 |
| E2F | 0.773 | 0.699 | 0.699 | 0.738 | 0.775 | 0.732 | 0.650 | 0.642 | 0.680 | 0.749 |
| ETS1 | 0.761 | 0.713 | 0.713 | 0.757 | 0.769 | 0.734 | 0.622 | 0.722 | 0.706 | 0.722 |
| AP-2gamma | 0.760 | 0.682 | 0.682 | 0.759 | 0.736 | 0.704 | 0.575 | 0.655 | 0.695 | 0.715 |
| MEF2A | 0.731 | 0.704 | 0.704 | 0.704 | 0.742 | 0.706 | 0.588 | 0.678 | 0.696 | 0.726 |
| RFX5 | 0.703 | 0.630 | 0.630 | 0.695 | 0.716 | 0.660 | 0.571 | 0.674 | 0.669 | 0.692 |
| FOXA2 | 0.710 | 0.662 | 0.662 | 0.706 | 0.714 | 0.681 | 0.596 | 0.693 | 0.686 | 0.659 |
| NFKB | 0.694 | 0.632 | 0.632 | 0.690 | 0.697 | 0.651 | 0.557 | 0.613 | 0.650 | 0.682 |
| SREBP1 | 0.647 | 0.567 | 0.567 | 0.623 | 0.638 | 0.603 | 0.521 | 0.587 | 0.514 | 0.593 |
| ELF1 | 0.689 | 0.616 | 0.616 | 0.682 | 0.690 | 0.631 | 0.570 | 0.597 | 0.622 | 0.654 |
| TAL1 | 0.673 | 0.615 | 0.615 | 0.670 | 0.689 | 0.652 | 0.531 | 0.603 | 0.645 | 0.680 |
| YY1 | 0.676 | 0.630 | 0.630 | 0.670 | 0.688 | 0.634 | 0.566 | 0.641 | 0.650 | 0.657 |
| TBP | 0.678 | 0.641 | 0.641 | 0.675 | 0.683 | 0.659 | 0.569 | 0.620 | 0.612 | 0.633 |
| POU2F2 | 0.659 | 0.614 | 0.614 | 0.656 | 0.684 | 0.636 | 0.556 | 0.606 | 0.600 | 0.623 |
| TCF4 | 0.672 | 0.629 | 0.629 | 0.668 | 0.671 | 0.633 | 0.548 | 0.616 | 0.642 | 0.655 |
| Oct-2 | 0.647 | 0.603 | 0.603 | 0.644 | 0.665 | 0.624 | 0.554 | 0.605 | 0.595 | 0.620 |
| GATA-2 | 0.661 | 0.578 | 0.578 | 0.660 | 0.658 | 0.606 | 0.541 | 0.601 | 0.618 | 0.634 |
| HNF4 | 0.656 | 0.575 | 0.575 | 0.650 | 0.657 | 0.609 | 0.542 | 0.595 | 0.625 | 0.639 |
| Eralpha | 0.639 | 0.598 | 0.598 | 0.637 | 0.660 | 0.621 | 0.548 | 0.601 | 0.632 | 0.645 |
| GR | 0.635 | 0.567 | 0.567 | 0.619 | 0.636 | 0.606 | 0.548 | 0.602 | 0.604 | 0.624 |
| Pbx3 | 0.594 | 0.616 | 0.616 | 0.566 | 0.657 | 0.608 | 0.558 | 0.528 | 0.564 | 0.570 |
| USF | 0.639 | 0.550 | 0.550 | 0.638 | 0.640 | 0.597 | 0.533 | 0.596 | 0.616 | 0.636 |
| SRF | 0.635 | 0.580 | 0.580 | 0.628 | 0.644 | 0.601 | 0.567 | 0.547 | 0.590 | 0.641 |
| IRF1 | 0.612 | 0.586 | 0.586 | 0.609 | 0.631 | 0.573 | 0.503 | 0.581 | 0.603 | 0.621 |
| PU.1 | 0.622 | 0.559 | 0.559 | 0.621 | 0.620 | 0.570 | 0.517 | 0.575 | 0.607 | 0.613 |
| GATA3 | 0.616 | 0.567 | 0.567 | 0.613 | 0.624 | 0.562 | 0.512 | 0.591 | 0.594 | 0.591 |
| NF-E2 | 0.609 | 0.508 | 0.508 | 0.602 | 0.614 | 0.555 | 0.524 | 0.589 | 0.590 | 0.604 |
| PAX5 | 0.561 | 0.538 | 0.538 | 0.554 | 0.598 | 0.566 | 0.542 | 0.587 | 0.605 | 0.607 |
| GATA-1 | 0.587 | 0.552 | 0.552 | 0.585 | 0.601 | 0.564 | 0.526 | 0.542 | 0.574 | 0.575 |
| EBF | 0.551 | 0.509 | 0.509 | 0.549 | 0.564 | 0.503 | 0.497 | 0.533 | 0.543 | 0.562 |

**Table 1. Cross-validation AUC accuracy of 10 ML algorithms (first row) in the prediction of TFBS for 38 different TF (first column) using the MS-ST dataset. For each TF, the highest AUC value is highlighted in red.**

### 2.2.2 Effect of Window Size on Prediction Accuracy

A key parameter in our predictor is the size of window surrounding the site under consideration that one considers. The choice of window sizes can lead to varying degrees of binding site information being captured. A window size that is too small will be very specific to a single position in the alignment and unable to detect nearby binding sites. On the other hand, a window size that is too large will capture distant binding sites that may be irrelevant and lead to noise in the data. We investigated the impact of the window size on the MS-ST predictor. Figure 24 shows the AUC accuracy of TFBS predictions in four different TF using different window sizes. We notice the same type of downward concave curves in all four TF where window sizes that are too large or too small results in lower accuracy. In fact, large window sizes can introduce enough noise such that the predictor can no longer find any correlation between TFBS concentration and functional TFBS. The maximum AUC accuracy appears to converge between windows of size of 100 to 600, which is interesting as this is similar to the average length of promoters and enhancers (100bp – 1000bp) (Sharan, 2007). Thus, to maximize the accuracy of our predictor, we use a window of size 450 in our experimentation, just slightly above the average maximum to capture more information regarding surrounding TFBS.

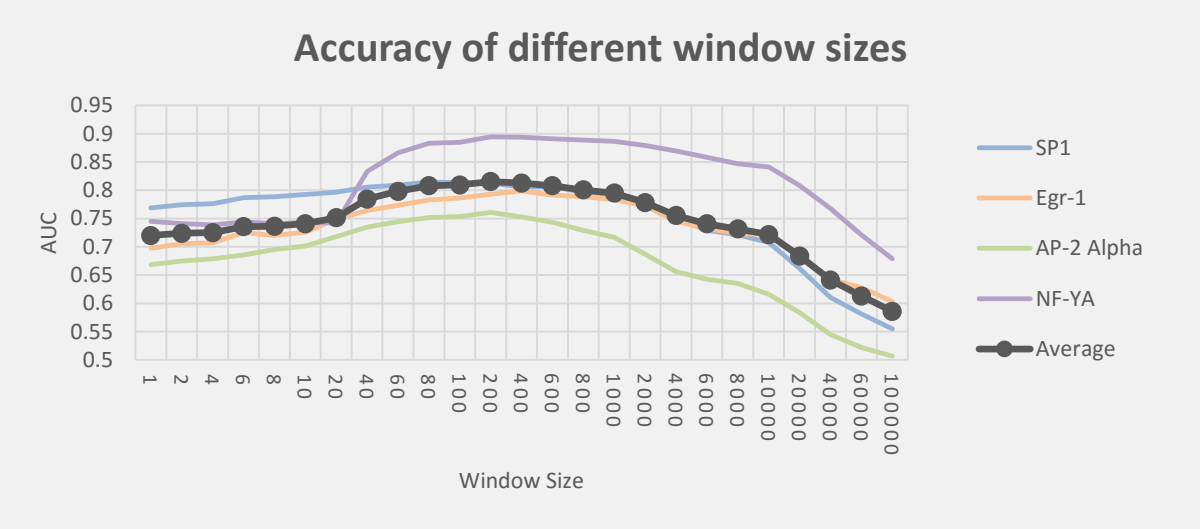



**Figure 24. Graph showing TFBS prediction accuracy of four different TFs given different window sizes.**

### 2.2.3 The Baseline predictor

Since the baseline predictor assumes that all examples presented to it are positive, its sensitivity is 100% and its specificity is 0%. The false-positive rate of this predictor is very high; it is by definition 50% on a balanced data set, but on an actual data set where negative examples far outnumber positive examples, the false-positive rate is much higher as shown in Figure 25, ranging from 54.5% false positives for TFs with highly specific PWMs (e.g. NRF1: 504 positive to 1111 total samples) to 99.9% false positives for those with short, information-poor TFs (e.g. SREBP1: 2624 positive to 1635981 total samples).

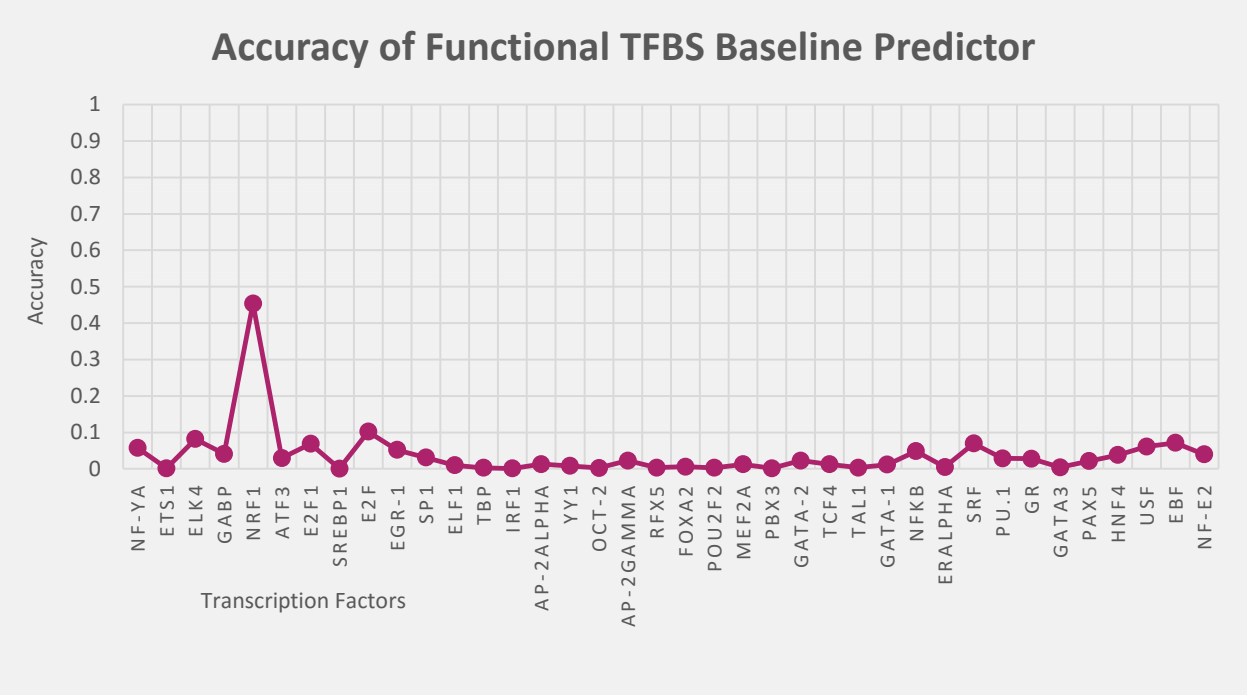



**Figure 25. Graph of the baseline predictor accuracy in the prediction of functional binding site of various TFs.**

### 2.2.4 The Single-species single-TF predictor

Figure 26 (orange line) reports the AUC of the SS-ST predictor on each of the 38 TFs considered. For most TFs, AUC values are barely above 0.5, which indicates that homotypic TFBS clustering is insufficient in itself to significantly improve functional TFBS predictions. In some cases, having redundant binding sites nearby can be an indication that the site is functional as TFBS turnover is a commonly observed

phenomenon in the genome (Dermitzakis & Clark, 2002). However, turnover is not a strict requirement for biological function and many TFBS do not have repeated sites nearby. A few exceptions are seen, for example, the Nuclear transcription factor Y subunit alpha (NF-YA) scored an AUC=0.82. This means that surrounding repeating binding sites plays an important role in the function of the TF.

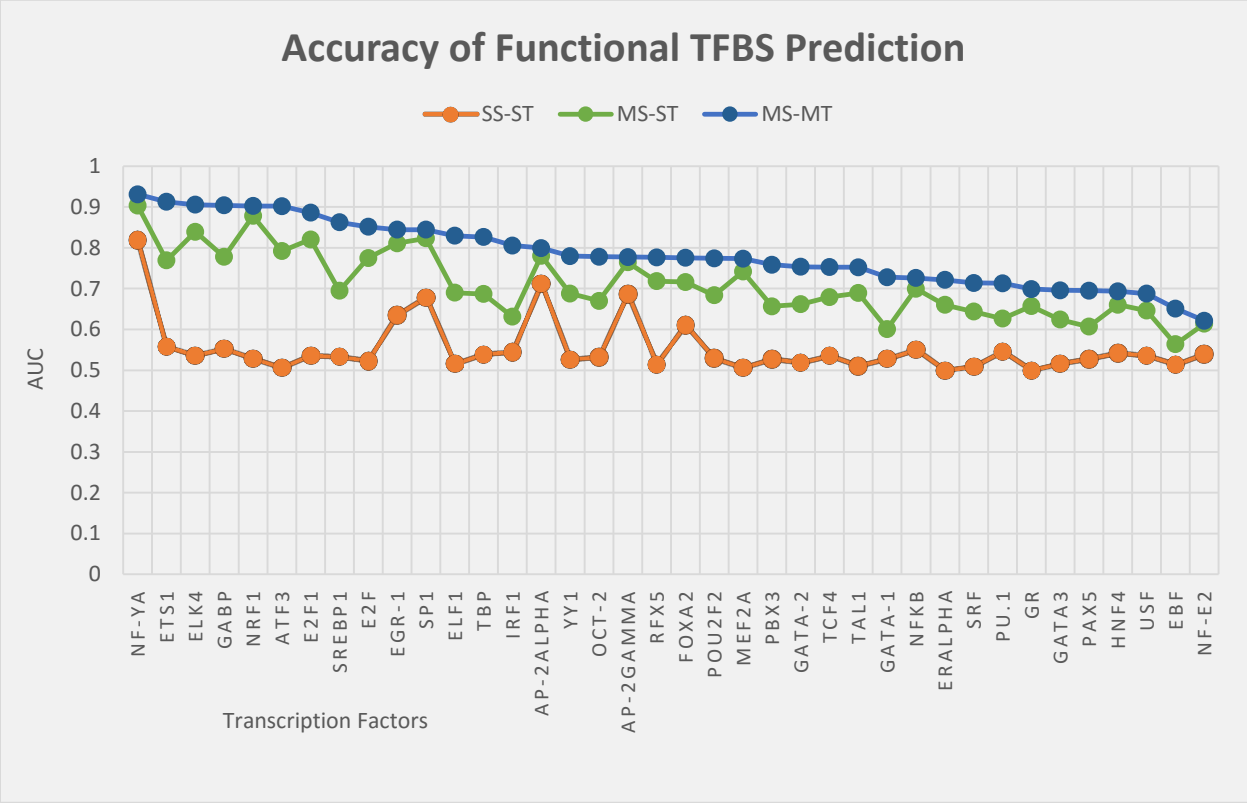



**Figure 26. Graph of the SVM AUC accuracy in the prediction of functional binding site for different TF on the SS-ST, MS-ST and MS-MT dataset.**

Analyzing the histogram of the number of PWM matches in the window surrounding positive and negative examples for NF-YA (Figure 27), we found that windows with more than two binding sites are highly likely to be functional, thus explaining the high accuracy. Similar observations are made for SP1, AP-2alpha and AP-2gamma. AP-2alpha and AP-2gamma are known to function as homodimer (National Center for Biotechnology Information, 2015). Biological reasons for the clustering of SP1 and NF-YA binding site requires further analysis.

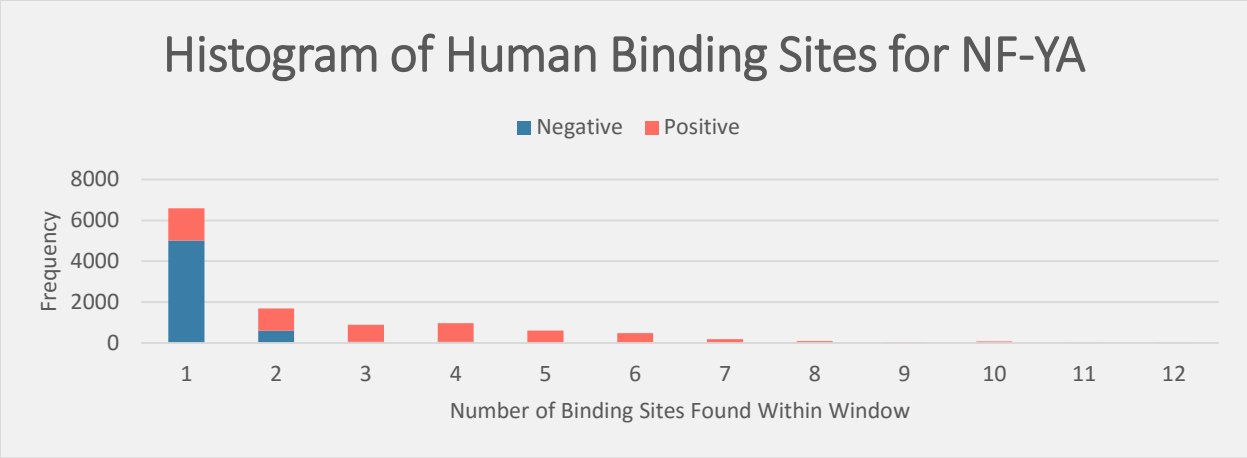



**Figure 27. A stacked histogram showing the frequency of bindings site found within each window. The color illustrates the proportion of the windows in each bin that are positive examples (orange) and negative examples (blue).**

### 2.2.5 The Multi-species Single-TF predictor

We then take the binding site data of other species and include them in our dataset. The addition of more species will allow us to capture more information regarding binding site conservation as functional TFBS experience selective pressure. TF that regulate genes that are orthologous in related species often leads to binding site conservation in the promoter of that gene. Thus, observing binding sites within a window in other related species might indicate a functional TFBS. As observed in Figure 26 (green curve), the MS-ST predictor significantly improves over the SS-ST predictor for all TFs considered, with AUC increases typically ranging from 10% to 25%, suggesting that the data of other species must be positively influencing the predictor. In fact, some TF such as NRF1 shows as much as a 35% increase in AUC, turning a very poor predictor (AUC=0.53) into an excellent one (AUC=0.88). In such cases, it is instructive to study the weights assigned by the logistic regression classifier to each the feature corresponding to each species and ancestor (Figure 28). Greater weights appear to have been given to primates and their ancestors (which are quite closely related to human) as well as hedgehog, hyrax and wallaby. Analysis of their histogram shows that the presence of inferred binding sites in these species tends to indicate biological function. The biological significance of why hyrax and wallaby have such strong impact on positive prediction is subject for further analysis.

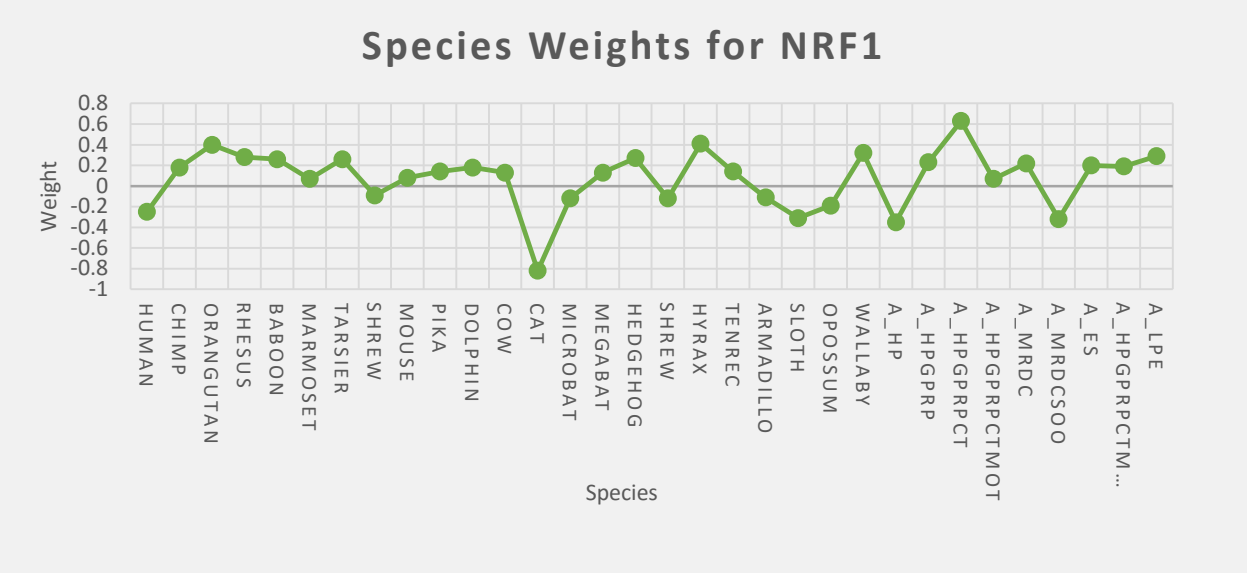



**Figure 28. Weights assigned to different species by the logistic regression classifier. "A_" represents an ancestral genome. Each letter after the underscore is the first character of the UCSC name given to the specie in which this is an ancestor of. For example, A_HP is the ancestor of human (HG) and chimp (PanTro). Species/Ancestors having a weight of zero is omitted.**

We also notice that cats, sloth and human all have negative weights, suggesting that a large concentration of predicted sites in these species is associated to non-functional PWM matches. Yet, contradiction is seen in their histogram where the presence of more inferred binding sites in these species still indicate function. What is common between their histogram is the lack of variance where most of their samples have the same binding site count. The reason for such apparent contradiction is that these species generally do not contribute much to the prediction since their variance is low. When the sample is positive, the negative weights in these species will be drown out by the numerous positive weights of other species. The negative weights comes into play when there is a negative sample yet a few species (including the species with negative weights) show presence of inferred binding site within the window. Since these negative-weight species are generally not significant, a strong negative weight can be used in this case to drown out the few positive-weights species. Nevertheless, it is quite possible that these negative samples are actually biologically functional in cell lines that are not included by the ENCODE data. Analyzing the histogram of all species and the logistic model equation, the trend appears

to be that the presence of more surrounding inferred binding sites tends to signify a functional TFBS. The negative weights could be there for the learner to counteract outlier negative samples.

### 2.2.6 The Multi-species Multi-TF Predictor

Some TF do not work alone and can only function when other TF binds nearby. In order to identify binding sites for TF *T*, the multi-species multi-TF predictor considers predicted binding sites for all 38 TFs jointly. This allows the predictor to learn how the existence of nearby binding sites for other TF affects the likelihood of functional sites. Although this comes with it a massive increase in the number of features, from 69 to 2622, it results in a significant improvement in AUC for many transcription factors (Figure 26, blue curve). Our results show that using the additional TF data can achieve as much as 15% increase in AUC accuracy. In particular, the fact that there is not a decrease in accuracy for any TF is a strong indication that the additional features provide relevant biological information to the learner. Having a large number of irrelevant (or noisy) features often results in over fitting as algorithms tries to fit these features to the output prediction. Remarkably, going from the MS-ST to the MS-MT feature set resulted in very large AUC increases for certain TFs. ETS1, for example, went from 56% to 91% AUC accuracy as a majority of its functional binding sites show high conservation in other species are located around clusters of TFBS.

### 2.2.7 Biological Significance of our Results

To further understand how the binding sites of other TF influence our prediction, we use WEKA's Simple Logistic learning algorithm (maximum 500 iterations for LogitBoost boosting, enabled heuristics and no weight trimming) to analyze our MS-MT dataset and examine the weights assigned to each TF in the prediction of other TF. Since we have a feature size of 2622, to interpret the result more clearly we first reduce the dimension by summing the weights of a given TF across all species/ancestor. The result is a matrix of weights of each TF's contribution on the prediction of every other TF. We show this matrix as a

heat map in Figure 29. As expected, the diagonal entries generally have large positive weights for TF in the prediction of itself. We also notice that a few TF tend to show stronger influence in the prediction of other TF's binding site. In particular, the presence of E2F1 and NF-E2 seems to have a strong impact on the positive prediction of numerous TF (as indicated by the high weights in their column). We focus on E2F1 as it influences the TFBS predictions of higher accuracy TF.

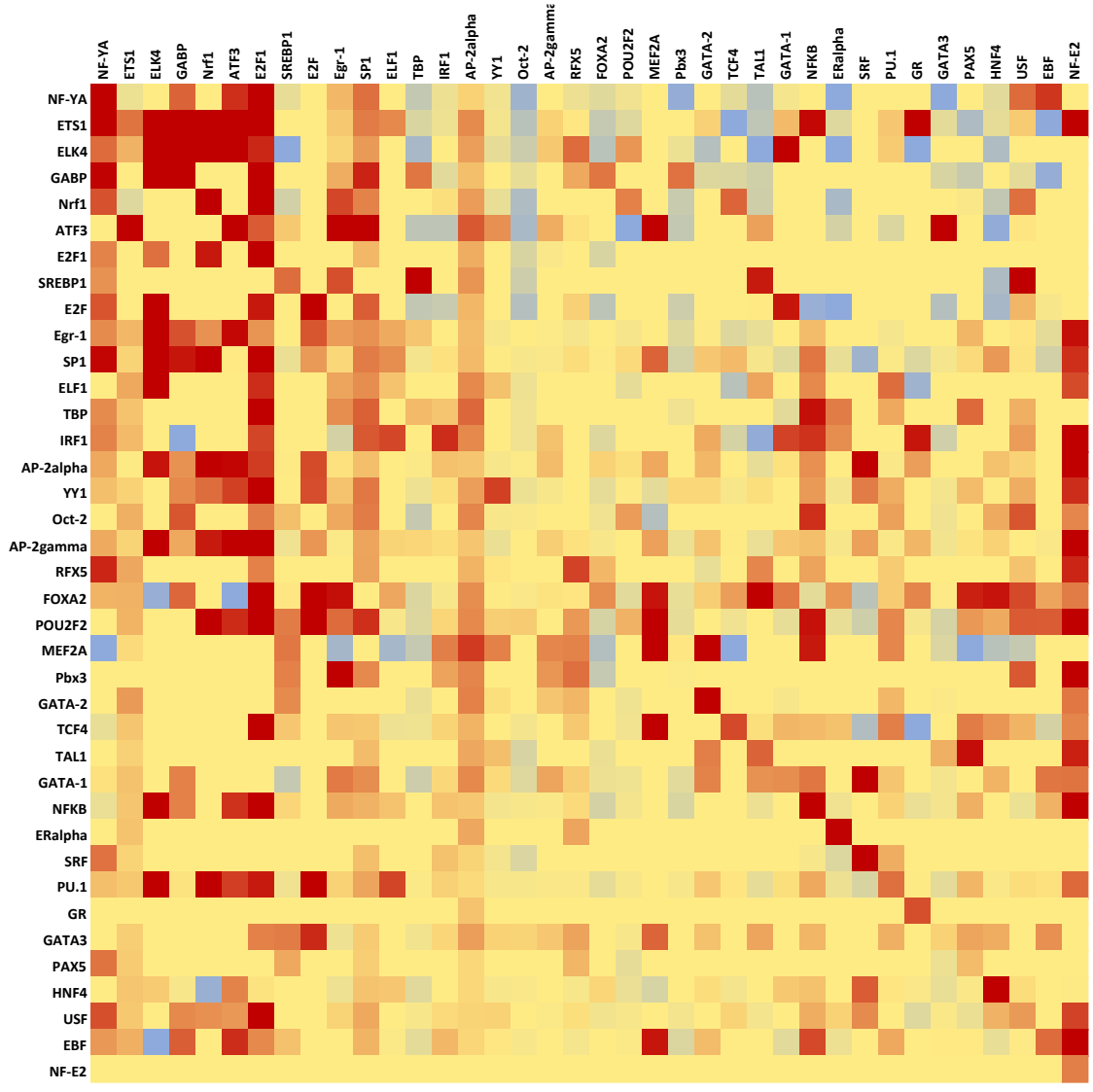



**Figure 29. Heatmap of the weights assigned by Simple Logistic on the MS-MT dataset. Each cell represent the cumulative weight assigned to the column TF on the prediction of the row TF's binding site. Red, yellow and blue indicate high, zero and negative weight. Heat map is sorted with the most accurately predicted TF on the top left.**

By analyzing the PWM of E2F1 as shown in Figure 2, we notice the strong presence of two CpG dinucleotides, which are very rare outside CpG islands. We wondered whether the presence of E2F1 sites in the window would actually be an indication that the window overlaps a CpG island, which are typically associated to the genes' promoter.

### 2.2.8 The role of CpG islands

CpG islands are regions in the genome where consecutive reoccurrence of alternating cytosine and guanine nucleotide is observed. Many mammalian DNA have CpG islands associated with promoters (approx. 70% of human promoters have high CpG content) (Reece, 2015). Since some TFs have very strong preference for binding promoters rather than enhancers, it may be that the large weights assigned to E2F1 features (and those of other TFs with GC-rich binding sites) may not suggest an association between the TFs and E2F1, but simply the fact that these TFs preferentially bind promoters. To test this, we separate the data into binding sites that are in CpG islands and those that are not. We then run Simple Logistic to re-compute the weights. The resulting heat map is shown in Figure 30 and Figure 30 for using samples in CpG islands only and Non-CpG islands only respectively (since the number of sites found in CpG islands are low, TF with samples size lower than 100 are omitted). We see that in both cases, TF continues to have positive prediction weights for itself as seen by the diagonal values in the heat map. But, when the data is separated, the coloring of the E2F1 column is much lighter indicating it no longer plays as important a role as it did before when looking at the CpG only heat map. The absence of E2F1 weights in many TF is even clearer for samples in Non-CpG islands. This seems to indicate the presence of E2F1 is less likely to be the result of interaction.

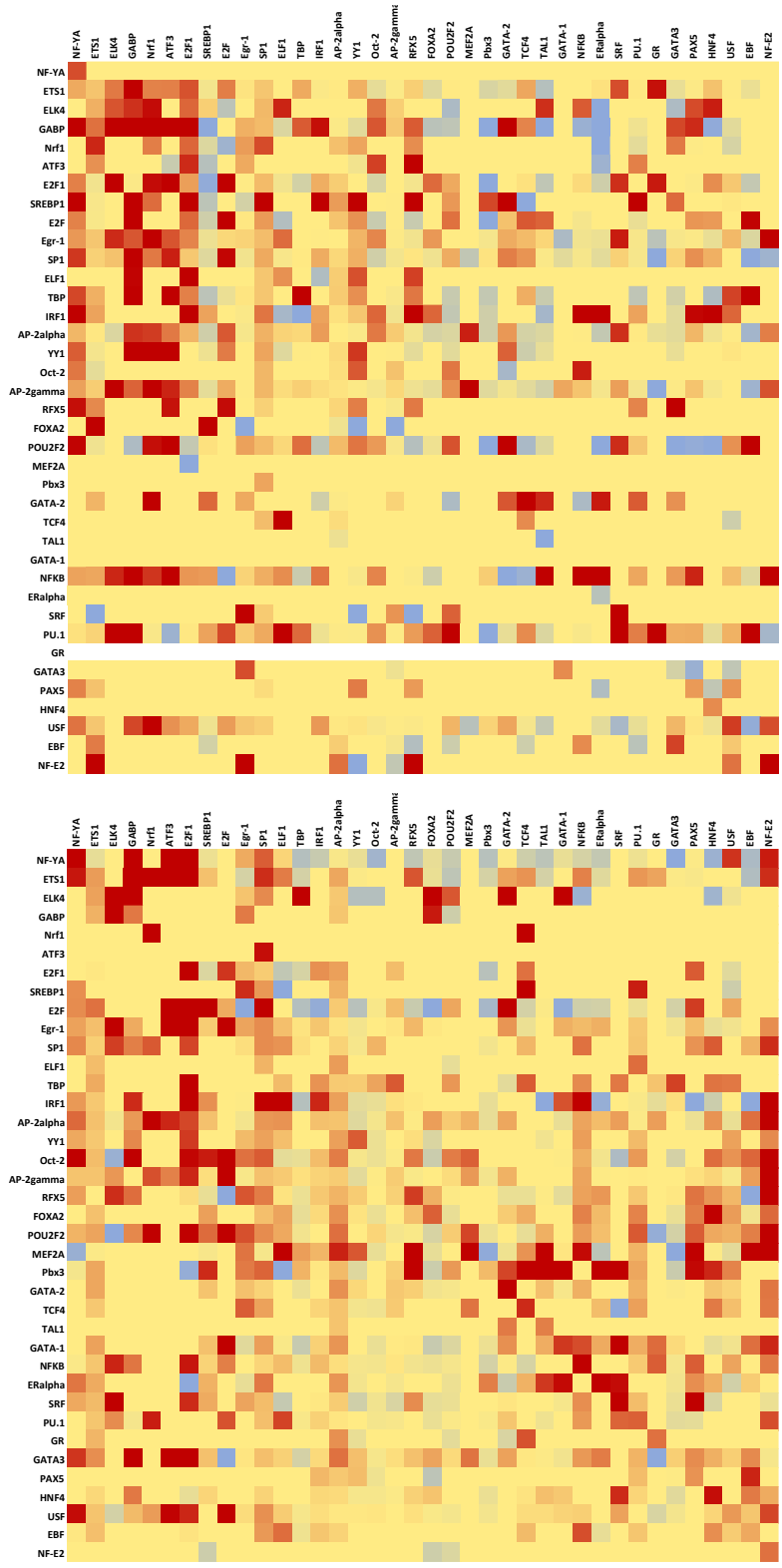



**Figure 30. Heat map of the weights assigned by Simple Logistic on the MS-MT dataset. CpG island samples only (top) and non-CpG island samples only (bottom).**

Some interesting observation can also be made regarding NF-YA. When the samples are limited to CpG islands only, this TF is not influenced by the presence of any other TF. This may be due to the TF's binding preference of 'CCAAT', which is a common promoter element (Fleming et al., 2013). As noted earlier, CpG islands are commonly associated with promoters and well-conserved binding site in promoters is sufficient for positive predictions.

The heatmap for CpG island samples only are also useful in identifying TFBS on the same promoter. High weights often indicate strong likelihood of functional binding site given the presence of the column TF. The highest weight in assigned in the CpG island only heat map is between NF-E2 and EGR-1, which is interesting as research have shown both TF are involved in the regulation of human thromboxane A2 receptor gene (Gannon & Kinsella, 2008).

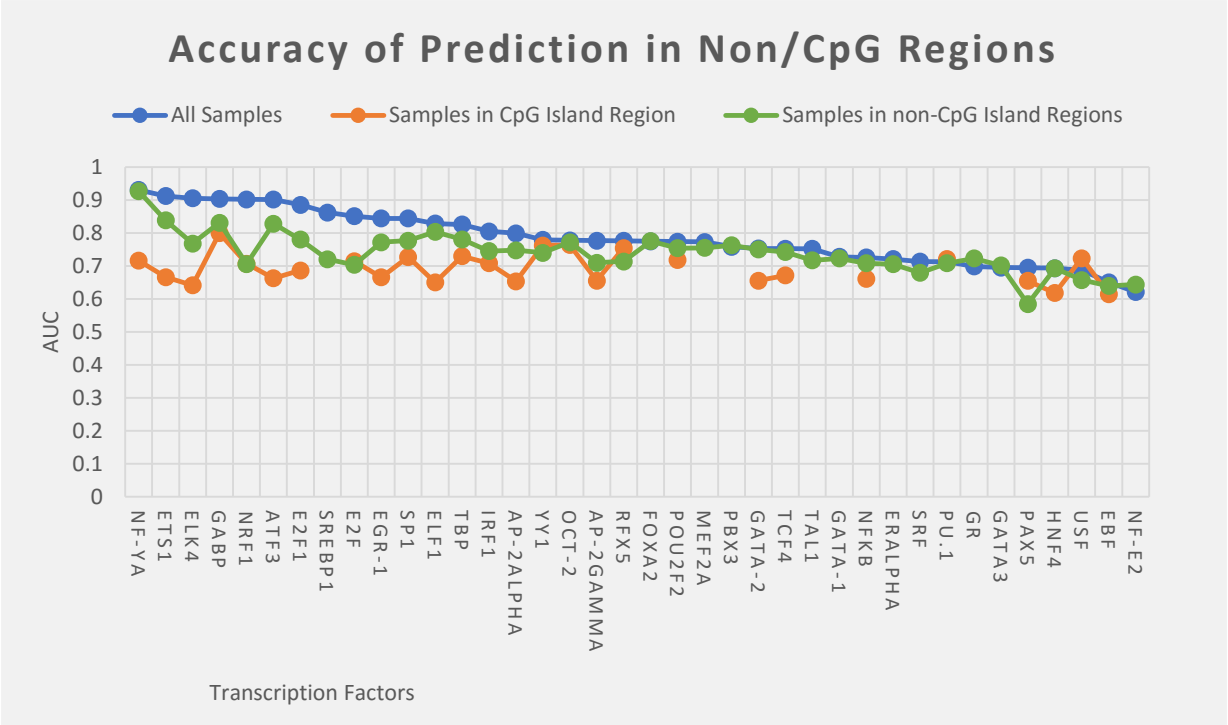



**Figure 31. AUC accuracy of all specie and all TF dataset. (Blue) indicate using all samples. (Orange) uses only samples found in CpG island regions. (Green) uses only samples found in non-CpG island regions.**

Looking at the learning accuracy for the CpG island separated dataset, we notice a decrease in accuracy for many of the TFBS (Figure 31). This means that the function of these TF has a strong dependence on whether or not they are within CpG enriched regions. For the other TF whose accuracy did not vary as drastically, they depend less on whether they are within CpG regions and more on the presence of surrounding binding sites.

Our results shows that analyzing the binding site count within a given window size works very well for many TFs. The addition of TFBS information on other species and the inclusion of binding sites data of multiple different TF have greatly improved the TFBS prediction accuracy for many of the TF we analyzed. However, our method does not work for all TF; deeper understanding of the biology of DNA-TF interaction is required for these TF such that more information can be captured to make better predictions. We believe our result is a strong demonstration of the ability of computational methods to model what we know about TFBS biology and positively predict the locations of functional TFBS on the genome. As TF binding behavior continues to be an on-going research, we are likely to see more improvements on computational predictors in the future.

# Chapter 3: Conclusion

Since the discovery of transcription factors, much research has been done to understand their function and their role in the gene regulatory network. An important step in this research is the identification of the transcription factor's binding sites on the genome. TFBS are often located near the genes they regulate and clustering of TFBS that belongs to different TF may indicate interaction between them. Learning about such interactions can be an important stepping-stone in our understanding of transcription regulation. Experimental methods, such as ChIP-Seq, is now frequently used by researchers to identify protein-DNA interactions *in vivo* and can be used to perform genome-wide mapping of TFBS. However, ChIP-Seq experiments are time consuming and expensive, as each experiment can take up to weeks to run and requires large amounts of starting material and access to sophisticated equipment.

With recent advances in sequencing technologies, such is Illumina, more and more species are being sequenced and their genomes are readily available. As a result, there is a growing interest in the use of genomic data to develop cost effective computational methods to identify the locations of TFBS on the genome. Experimentally determined TF motifs, in the form of Position Weight Matrix, are often used to scan the genome for sequence matches; however, such method suffers from the problem of false positives predictions. Although motif matches on the genome may indicate a legitimate binding site, many of these sites are never used since many of the predicted binding sites are either a motif match purely by chance or wound up by histones into nucleosomes and are inaccessible to TF. Numerous amount of research has been done to filter out the false positive sites and improve the overall accuracy.

Our research is motivated by the rapidly growing field of machine learning and how its ability for pattern recognition can improve the filtering of false positive binding site predictions. By performing multiple alignment of orthologous sequences from multiple species and their ancestral sequences and transforming the binding site count into a ML dataset, we want to observe whether a machine learning

algorithm, such as a SVM, can identify if a PWM inferred binding site in human is biologically functional based on the presence of other TFBS belonging to different TF in multiple other species and ancestral genomes.

Our results show such method is extremely effective on certain TF, such as ETS1, having an AUC accuracy as high as 91%. However, it should be noted that not all TF's binding site predictions could achieve the same accuracy gain by using this method. Only one-third of the 38 TF we tested on achieved an overall AUC accuracy > 80%; this is not surprising as we recognize there are numerous biological principles that leads to *in vivo* TF binding that our dataset does not capture. There is unlikely a catchall method to accurately predict all TFBS equally well. Functional TFBS that tend to form clusters with other TFBS and have strong evolutionary conservation show the best accuracy with our method.

Many factors limit the performance of our learner. First, more than 200 known cell lines exists in human; however, the ENCODE project only performed ChIP-Seq experiment on approx. 100 human cell line which are divided into 3 different tiers where tier 1 and 2 cell lines (totaling of 6 cell lines) are extensively studied with multiple TF while Tier 3 are studied with fewer TF (Consortium, 2012). This means that there could be more regions in the genome that are actually functional binding sites but have not yet been identified experimentally. Secondly, a 2004 paper showed that there are around 1850 known TF in humans (Bulyk, 2003) with more predicted to be verified in the future. Our experiments only analyze 38 of them, as they were the only TF in which we have both the ChIP-Seq and PWM data for. It is possible the addition of more TF can further improve the accuracy of our learner as more information regard TF-TF interactions and TF clustering can be captured.

Given that prediction of functional TFBS and filtering out of false positive binding sites has been the focus of ongoing research efforts for many years, numerous research papers and tools exist that addresses this problem. Existing tools such as rVista and CONTRASIF uses genome of multiple species to

detect phylogenetic conservation of TFBS as a way to identify true binding sites. Other papers, such as those by Ramsey et al. and Xu et al., uses genome wide acetylation and methylation data in their TFBS prediction as acetylation is often associated with open chromatin (Ramsey et al., 2010) and methylation of cytosine is known to interfere with protein-DNA interaction (Xu et al., 2015). All these are valuable data that could be incorporated into our predictor to improve its performance.

Moreover, it is expected that comparison of sequences from multiple species, especially more distance species, will lead to significant refinement in our understanding of the functional regions in our genome (Collins et al., 2003). In our analysis, we use only 35 mammalian species. As more and more species are being sequenced, prediction accuracy could be improved with the addition of more distal species genome.

We hope that our work will serve as a stepping-stone for future research in the understanding of TF and gene regulation. We have demonstrated that the combined analysis of multi-species TFBS conservation and TF clustering is a powerful way to filter out non-functional binding sites. We also show that machine learning can be an invaluable tool, not only in the development of binding site predictors, but by analyzing the mathematical model learnt from the binding site dataset, much can be learnt regarding TF-TF interactions and the binding preference of TF.

This research is by no means a conclusive work. We have shown one novel method of functional TFBS prediction using machine learning and interspecies comparison. It is clear that computational methods can be effective and has much to offer; however, much work still remains in the development of better predictive model as we continue to learn more about the binding criterion between TF and TFBS.

# References


Alberts, B. (2007). Essential cell biology. Princeton, NJ: Recording for the Blind & Dyslexic.

Babu, M. M., Luscombe, N. M., Aravind, L., Gerstein, M., & Teichmann, S. A. (2004). Structure and evolution of transcriptional regulatory networks. *Curr Opin Struct Biol, 14*(3), 283-291. doi:10.1016/j.sbi.2004.05.004

Baldwin, A. S., Jr. (2001). Series introduction: the transcription factor NF-kappaB and human disease. *J Clin Invest, 107*(1), 3-6. doi:10.1172/JCI11891

Beato, M., & Eisfeld, K. (1997). Transcription factor access to chromatin. *Nucleic Acids Res, 25*(18), 3559-3563. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/9278473

Behnel, S., Bradshaw, R., Citro, C., Dalcin, L., Seljebotn, D. S., & Smith, K. (2011). Cython: The best of both worlds. *Computing in Science & Engineering, 13*(2), 31-39.

Besse, F., & Ephrussi, A. (2008). Translational control of localized mRNAs: restricting protein synthesis in space and time. *Nature reviews Molecular cell biology, 9*(12), 971-980.

Blanchette, M. (2012). Exploiting ancestral mammalian genomes for the prediction of human transcription factor binding sites. *BMC Bioinformatics, 13 Suppl 19*, S2. doi:10.1186/1471-2105-13-S19-S2

Blanchette, M., Bataille, A. R., Chen, X., Poitras, C., Laganiere, J., Lefebvre, C., . . . Robert, F. (2006). Genome-wide computational prediction of transcriptional regulatory modules reveals new insights into human gene expression. *Genome Res, 16*(5), 656-668. doi:10.1101/gr.4866006

Blanchette, M., Green, E. D., Miller, W., & Haussler, D. (2004). Reconstructing large regions of an ancestral mammalian genome in silico. *Genome Res, 14*(12), 2412-2423. doi:10.1101/gr.2800104

Blanchette, M., Kent, W. J., Riemer, C., Elnitski, L., Smit, A. F., Roskin, K. M., . . . Miller, W. (2004). Aligning multiple genomic sequences with the threaded blockset aligner. *Genome Res, 14*(4), 708-715. doi:10.1101/gr.1933104

Blanchette, M., & Tompa, M. (2002). Discovery of regulatory elements by a computational method for phylogenetic footprinting. *Genome Res, 12*(5), 739-748. doi:10.1101/gr.6902

Bock, C., Walter, J., Paulsen, M., & Lengauer, T. (2007). CpG island mapping by epigenome prediction. *PLoS Comput Biol, 3*(6), e110. doi:10.1371/journal.pcbi.0030110

Bonifacino, J. S., Dell'Angelica, E. C., & Springer, T. A. (2001). Immunoprecipitation. *Curr Protoc Immunol, Chapter 8*, Unit 8 3. doi:10.1002/0471142735.im0803s41

Buck, M. J., & Lieb, J. D. (2004). ChIP-chip: considerations for the design, analysis, and application of genome-wide chromatin immunoprecipitation experiments. *Genomics, 83*(3), 349-360. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/14986705

Bulyk, M. L. (2003). Computational prediction of transcription-factor binding site locations. *Genome Biol, 5*(1), 201. doi:10.1186/gb-2003-5-1-201

Carey, M. F., Peterson, C. L., & Smale, S. T. (2009). Chromatin immunoprecipitation (ChIP). *Cold Spring Harbor Protocols, 2009*(9), pdb. prot5279. Retrieved from http://cshprotocols.cshlp.org/content/2009/9/pdb.prot5279.full.pdf

Carmody, S. R., & Wente, S. R. (2009). mRNA nuclear export at a glance. *Journal of Cell Science, 122*(12), 1933-1937.

Caruana, R., & Niculescu-Mizil, A. (2006). *An empirical comparison of supervised learning algorithms.* Paper presented at the Proceedings of the 23rd international conference on Machine learning.

Chang, C.-C., & Lin, C.-J. (2011). LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol., 2*(3), 1-27. doi:10.1145/1961189.1961199

Chatfield, C. (1991). Avoiding statistical pitfalls. *Statistical Science, 6*(3), 240-252.

Claussnitzer, M., Dankel, S. N., Klocke, B., Grallert, H., Glunk, V., Berulava, T., . . . Laumen, H. (2014). Leveraging cross-species transcription factor binding site patterns: from diabetes risk loci to disease mechanisms. *Cell, 156*(1-2), 343-358. doi:10.1016/j.cell.2013.10.058

Collins, F. S., Green, E. D., Guttmacher, A. E., & Guyer, M. S. (2003). A vision for the future of genomics research. *Nature, 422*(6934), 835-847. doi:10.1038/nature01626

Consortium, E. P. (2012). An integrated encyclopedia of DNA elements in the human genome. *Nature, 489*(7414), 57-74. doi:10.1038/nature11247

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning, 20*(3), 273-297.

da Piedade, I., Tang, M. H., & Elemento, O. (2009). DISPARE: DIScriminative PAttern REfinement for Position Weight Matrices. *BMC Bioinformatics, 10*, 388. doi:10.1186/1471-2105-10-388

Day, D. A., & Tuite, M. F. (1998). Post-transcriptional gene regulatory mechanisms in eukaryotes: an overview. *J Endocrinol, 157*(3), 361-371. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/9691970

Deaton, A. M., & Bird, A. (2011). CpG islands and the regulation of transcription. *Genes Dev, 25*(10), 1010-1022. doi:10.1101/gad.2037511

Dermitzakis, E. T., & Clark, A. G. (2002). Evolution of transcription factor binding sites in Mammalian gene regulatory regions: conservation and turnover. *Mol Biol Evol, 19*(7), 1114-1121. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/12082130

Djordjevic, M. (2007). SELEX experiments: new prospects, applications and data analysis in inferring regulatory pathways. *Biomol Eng, 24*(2), 179-189. doi:10.1016/j.bioeng.2007.03.001

Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM, 55*(10), 78-87.

Elati, M., & Rouveirol, C. (2010). Unsupervised learning for gene regulation network inference from expression data: A review. *Algorithms in Computational Molecular Biology: Techniques, Approaches and Applications*, 955-978.

Epigentek. (2015). EpiNext ChIP-Seq High-Sensitivity Kit (Illumina). Retrieved from http://www.epigentek.com/catalog/epinext-chip-seq-high-sensitivity-kit-illumina-p-3650.html?currency=ca&height=190&width=500&border=1&modal=true&random=1436279059946

Ferretti, V., Poitras, C., Bergeron, D., Coulombe, B., Robert, F., & Blanchette, M. (2007). PReMod: a database of genome-wide mammalian cis-regulatory module predictions. *Nucleic Acids Res, 35*(Database issue), D122-126. doi:10.1093/nar/gkl879

Fleming, J. D., Pavesi, G., Benatti, P., Imbriano, C., Mantovani, R., & Struhl, K. (2013). NF-Y coassociates with FOS at promoters, enhancers, repetitive elements, and inactive chromatin regions, and is stereo-positioned with growth-controlling transcription factors. *Genome Res, 23*(8), 1195-1209. doi:10.1101/gr.148080.112

Frietze, S., & Farnham, P. J. (2011). Transcription factor effector domains. *Subcell Biochem, 52*, 261-277. doi:10.1007/978-90-481-9069-0_12

Ganley, A. R., & Kobayashi, T. (2007). Phylogenetic footprinting to find functional DNA elements. *Methods Mol Biol, 395*, 367-380. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/17993686

Gannon, A. M., & Kinsella, B. T. (2008). Regulation of the human thromboxane A2 receptor gene by Sp1, Egr1, NF-E2, GATA-1, and Ets-1 in megakaryocytes. *J Lipid Res, 49*(12), 2590-2604. doi:10.1194/jlr.M800256-JLR200

Geertz, M., & Maerkl, S. J. (2010). Experimental strategies for studying transcription factor-DNA binding specificities. *Brief Funct Genomics, 9*(5-6), 362-373. doi:10.1093/bfgp/elq023

Goldberg, D. (1989). Genetic algorithms in search, optimization, and machine learning.

Guindon, S., & Gascuel, O. (2003). A simple, fast, and accurate algorithm to estimate large phylogenies by maximum likelihood. *Syst Biol, 52*(5), 696-704. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/14530136

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: an update. *ACM SIGKDD explorations newsletter, 11*(1), 10-18.

Hardison, R. C. (2003). Comparative genomics. *PLoS Biol, 1*(2), E58. doi:10.1371/journal.pbio.0000058

Hasegawa, M., Kishino, H., & Yano, T. (1985). Dating of the human-ape splitting by a molecular clock of mitochondrial DNA. *J Mol Evol, 22*(2), 160-174. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/3934395

Holloway, D. T., Kon, M., & DeLisi, C. (2005). Integrating genomic data to predict transcription factor binding. *Genome Inform, 16*(1), 83-94. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/16362910

Jeff Hardin, G. P. B., Lewis J. Kleinsmith. (2012). *Becker's World of the Cell* (8th ed.): Boston : Benjamin Cummings, 2012.

Johnson, J. (2013). General regression and over fitting. *The Shape of Data.* Retrieved from https://shapeofdata.wordpress.com/2013/03/26/general-regression-and-over-fitting/

Kamanu, F. K., Medvedeva, Y. A., Schaefer, U., Jankovic, B. R., Archer, J. A., & Bajic, V. B. (2012). Mutations and binding sites of human transcription factors. *Front Genet, 3*, 100. doi:10.3389/fgene.2012.00100

Kel, A. E., Gossling, E., Reuter, I., Cheremushkin, E., Kel-Margoulis, O. V., & Wingender, E. (2003). MATCH: A tool for searching transcription factor binding sites in DNA sequences. *Nucleic Acids Res, 31*(13), 3576-3579. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/12824369

Kim, E. (2013). Everything You Wanted to Know about the Kernel Trick (But Were Too Afraid to Ask). Retrieved from http://www.eric-kim.net/eric-kim-net/posts/1/kernel_trick.html

Kohavi, R. (1995). *A study of cross-validation and bootstrap for accuracy estimation and model selection*.

Kohler, A., & Hurt, E. (2007). Exporting RNA from the nucleus to the cytoplasm. *Nat Rev Mol Cell Biol, 8*(10), 761-773. doi:10.1038/nrm2255

Korhonen, J., Martinmaki, P., Pizzi, C., Rastas, P., & Ukkonen, E. (2009). MOODS: fast search for position weight matrix matches in DNA sequences. *Bioinformatics, 25*(23), 3181-3182. doi:10.1093/bioinformatics/btp554

Kullback, S., & Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 79-86.

Landt, S. G., Marinov, G. K., Kundaje, A., Kheradpour, P., Pauli, F., Batzoglou, S., . . . Snyder, M. (2012). ChIP-seq guidelines and practices of the ENCODE and modENCODE consortia. *Genome Res, 22*(9), 1813-1831. doi:10.1101/gr.136184.111

Latchman, D. S. (1997). Transcription factors: an overview. *Int J Biochem Cell Biol, 29*(12), 1305-1312. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/9570129

Lawrence, C. E., Altschul, S. F., Boguski, M. S., Liu, J. S., Neuwald, A. F., & Wootton, J. C. (1993). Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science, 262*(5131), 208-214. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/8211139

Li, L., Liang, Y., & Bass, R. L. (2007). GAPWM: a genetic algorithm method for optimizing a position weight matrix. *Bioinformatics, 23*(10), 1188-1194. doi:10.1093/bioinformatics/btm080

Lobo, I. (2008). Environmental influences on gene expression. *Nature Education, 1*(1), 39.

Loots, G. G., & Ovcharenko, I. (2004). rVISTA 2.0: evolutionary analysis of transcription factor binding sites. *Nucleic Acids Res, 32*(Web Server issue), W217-221. doi:10.1093/nar/gkh383

Maerkl, S. J., & Quake, S. R. (2007). A systems approach to measuring the binding energy landscapes of transcription factors. *Science, 315*(5809), 233-237. doi:10.1126/science.1131007

Maerkl, S. J., & Quake, S. R. (2009). Experimental determination of the evolvability of a transcription factor. *Proc Natl Acad Sci U S A, 106*(44), 18650-18655. doi:10.1073/pnas.0907688106

Marinescu, V. D., Kohane, I. S., & Riva, A. (2005). The MAPPER database: a multi-genome catalog of putative transcription factor binding sites. *Nucleic Acids Res, 33*(Database issue), D91-97. doi:10.1093/nar/gki103

Mathelier, A., & Wasserman, W. W. (2013). The next generation of transcription factor binding site prediction. *PLoS Comput Biol, 9*(9), e1003214. doi:10.1371/journal.pcbi.1003214

Matys, V., Fricke, E., Geffers, R., Gossling, E., Haubrock, M., Hehl, R., . . . Wingender, E. (2003). TRANSFAC: transcriptional regulation, from patterns to profiles. *Nucleic Acids Res, 31*(1), 374-378. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/12520026

McCrea, N. (2014). An Introduction to Machine Learning Theory and Its Applications: A Visual Tutorial with Examples. Retrieved from http://www.toptal.com/machine-learning/machine-learning-theory-an-introductory-primer

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics, 21*(6), 1087-1092.

Miller, W., Rosenbloom, K., Hardison, R. C., Hou, M., Taylor, J., Raney, B., . . . Kent, W. J. (2007). 28-way vertebrate alignment and conservation track in the UCSC Genome Browser. *Genome Res, 17*(12), 1797-1808. doi:10.1101/gr.6761107

Mitchell, T. M. (2006). *The discipline of machine learning* (Vol. 17).

National Center for Biotechnology Information, U. S. N. L. o. M. (2015). TFAP2C transcription factor AP-2 gamma. Retrieved from http://www.ncbi.nlm.nih.gov/gene/7022

NatureEducation. (2014a). Enhancer. Retrieved from http://www.nature.com/scitable/definition/enhancer-163

NatureEducation. (2014b). Promoter. Retrieved from http://www.nature.com/scitable/definition/promoter-259

Orenstein, Y., Linhart, C., & Shamir, R. (2012). Assessment of algorithms for inferring positional weight matrix motifs of transcription factor binding sites using protein binding microarray data.

Ovcharenko, I., Loots, G. G., Giardine, B. M., Hou, M., Ma, J., Hardison, R. C., . . . Miller, W. (2005). Mulan: multiple-sequence local alignment and visualization for studying function and evolution. *Genome Res, 15*(1), 184-194. doi:10.1101/gr.3007205

Park, P. J. (2009). ChIP-seq: advantages and challenges of a maturing technology. *Nat Rev Genet, 10*(10), 669-680. doi:10.1038/nrg2641

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. (2011). Scikit-learn: Machine learning in Python. *The Journal of Machine Learning Research, 12*, 2825-2830.

Pennacchio, L. A., & Rubin, E. M. (2001). Genomic strategies to identify mammalian regulatory sequences. *Nat Rev Genet, 2*(2), 100-109. doi:10.1038/35052548

Peters, T. (2002). Python-Dev Sorting. Retrieved from https://mail.python.org/pipermail/python-dev/2002-July/026837.html

Phillips, T. (2008). Regulation of transcription and gene expression in eukaryotes. *Nature Education, 1*(1), 199.

Pique-Regi, R., Degner, J. F., Pai, A. A., Gaffney, D. J., Gilad, Y., & Pritchard, J. K. (2011). Accurate inference of transcription factor binding from DNA sequence and chromatin accessibility data. *Genome Res, 21*(3), 447-455. doi:10.1101/gr.112623.110

Preiss, T. (2000). The end in sight: Poly (A), translation and mRNA stability in eukaryotes.

Qian, Z., Cai, Y.-D., & Li, Y. (2006). A novel computational method to predict transcription factor DNA binding preference. *Biochemical and biophysical research communications, 348*(3), 1034-1037. Retrieved from http://www.sciencedirect.com/science/article/pii/S0006291X06017207

Quaresma, A. J., Sievert, R., & Nickerson, J. A. (2013). Regulation of mRNA export by the PI3 kinase/AKT signal transduction pathway. *Mol Biol Cell, 24*(8), 1208-1221. doi:10.1091/mbc.E12-06-0450

Ramsey, S. A., Knijnenburg, T. A., Kennedy, K. A., Zak, D. E., Gilchrist, M., Gold, E. S., . . . Shmulevich, I. (2010). Genome-wide histone acetylation data improve prediction of mammalian transcription factor binding sites. *Bioinformatics, 26*(17), 2071-2075. doi:10.1093/bioinformatics/btq405

Reece, R. J. (2015). *Analysis of Genes and Genomes*: John Wiley & Sons, Limited.

Sadri, J., Diallo, A. B., & Blanchette, M. (2011). Predicting site-specific human selective pressure using evolutionary signatures. *Bioinformatics, 27*(13), i266-274. doi:10.1093/bioinformatics/btr241

Safavian, S. R., & Landgrebe, D. (1991). A survey of decision tree classifier methodology. *IEEE transactions on systems, man, and cybernetics, 21*(3), 660-674.

Sambrook, J., & Russell, D. W. (2006). Fragmentation of DNA by sonication. *CSH Protoc, 2006*(4). doi:10.1101/pdb.prot4538

Sandelin, A., Alkema, W., Engstrom, P., Wasserman, W. W., & Lenhard, B. (2004). JASPAR: an open-access database for eukaryotic transcription factor binding profiles. *Nucleic Acids Res, 32*(Database issue), D91-94. doi:10.1093/nar/gkh012

Schwartz, S., Kent, W. J., Smit, A., Zhang, Z., Baertsch, R., Hardison, R. C., . . . Miller, W. (2003). Human-mouse alignments with BLASTZ. *Genome Res, 13*(1), 103-107. doi:10.1101/gr.809403

scikit-learn. (2010-2011a). K-means clustering. *Unsupervised learning: seeking representations of the data.* Retrieved from http://scikit-learn.org/0.11/tutorial/statistical_inference/unsupervised_learning.html

scikit-learn. (2010-2011b). RBF SVM parameters. *SVM.* Retrieved from http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

scikit-learn. (2010-2011c). Receiver Operating Characteristic (ROC) with cross validation. *Receiver operating characteristic (ROC).* Retrieved from http://scikit-learn.org/stable/auto_examples/model_selection/plot_roc_crossval.html#example-model-selection-plot-roc-crossval-py

Sharan, R. (2007). Analysis of Biological Networks: Transcriptional Networks - Promoter Sequence Analysis. Retrieved from http://www.cs.tau.ac.il/~roded/courses/bnet-a06/lec11.pdf

Siddharthan, R. (2010). Dinucleotide weight matrices for predicting transcription factor binding sites: generalizing the position weight matrix. *PLoS One, 5*(3), e9722. doi:10.1371/journal.pone.0009722

Smith, C. M., Marks, A. D., Lieberman, M. A., & Marks, D. B. (2005). *Marks' Basic Medical Biochemistry: A Clinical Approach*: Lippincott Williams & Wilkins.

Song, K. (2012). Transcription Factors. In T. Factors.svg (Ed.), *Scalable Vector Graphics* (Vol. 1,572 × 853 pixels, pp. Diagram of gene transcription factors. Note—This image serves as its own editable text version—the editable text layer is invisible, underneath the outline text. Open it in Inkscape and follow the instructions outside the image boundary.): Wikipedia Commons.

Sprawls, P. (1995). Image Characteristics and Quality *Physical Principles of Medical Imaging* (2 ed., pp. 656): Medical Physics Publishing.

Stewart, A. J., Hannenhalli, S., & Plotkin, J. B. (2012). Why transcription factor binding sites are ten nucleotides long. *Genetics, 192*(3), 973-985. doi:10.1534/genetics.112.143370

Stormo, G. D., & Hartzell, G. W., 3rd. (1989). Identifying protein-binding sites from unaligned DNA fragments. *Proc Natl Acad Sci U S A, 86*(4), 1183-1187. Retrieved from http://www.ncbi.nlm.nih.gov/pubmed/2919167

Tang, D. (2013, October 1). Position Weight Matrix. *Musing From a PhD Candidate.* Retrieved from http://davetang.org/muse/2013/10/01/position-weight-matrix/

Tempel, S. (2012). Using and understanding RepeatMasker. *Methods Mol Biol, 859*, 29-51. doi:10.1007/978-1-61779-603-6_2

Thomas, M. C., & Chiang, C. M. (2006). The general transcription machinery and general cofactors. *Crit Rev Biochem Mol Biol, 41*(3), 105-178. doi:10.1080/10409230600648736

Tokovenko, B., Golda, R., Protas, O., Obolenskaya, M., & El'skaya, A. (2009). COTRASIF: conservation-aided transcription-factor-binding site finder. *Nucleic Acids Res, 37*(7), e49. doi:10.1093/nar/gkp084

Trevino, V., Falciani, F., & Barrera-Saldana, H. A. (2007). DNA microarrays: a powerful genomic tool for biomedical and clinical research. *Mol Med, 13*(9-10), 527-541. doi:10.2119/2006-00107.Trevino

van Straalen, N. M., & Roelofs, D. (2006). *An Introduction to Ecological Genomics*: Oxford University Press.

Wang, S.-C. (2003). *Interdisciplinary Computing in Java Programming* (Vol. 743): Springer US.

Xu, T., Li, B., Zhao, M., Szulwach, K. E., Street, R. C., Lin, L., . . . Qin, Z. S. (2015). Base-resolution methylation patterns accurately predict transcription factor bindings in vivo. *Nucleic Acids Res, 43*(5), 2757-2766. doi:10.1093/nar/gkv151

Yu, H., Luscombe, N. M., Qian, J., & Gerstein, M. (2003). Genomic analysis of gene expression relationships in transcriptional regulatory networks. *Trends Genet, 19*(8), 422-427. doi:10.1016/S0168-9525(03)00175-6

Ziliang Qian, Z. H., Yudong Cai. (2005). *Applying Machine Learning Strategy in Transcription Factor DNA Bindings Site Prediction*: iConcept Press.

Zykovich, A., Korf, I., & Segal, D. J. (2009). Bind-n-Seq: high-throughput analysis of in vitro protein-DNA interactions using massively parallel sequencing. *Nucleic Acids Res, 37*(22), e151. doi:10.1093/nar/gkp802

# Appendix

## A-1 Transcription Factors

ENCODE Transcription Factors That Were Examined

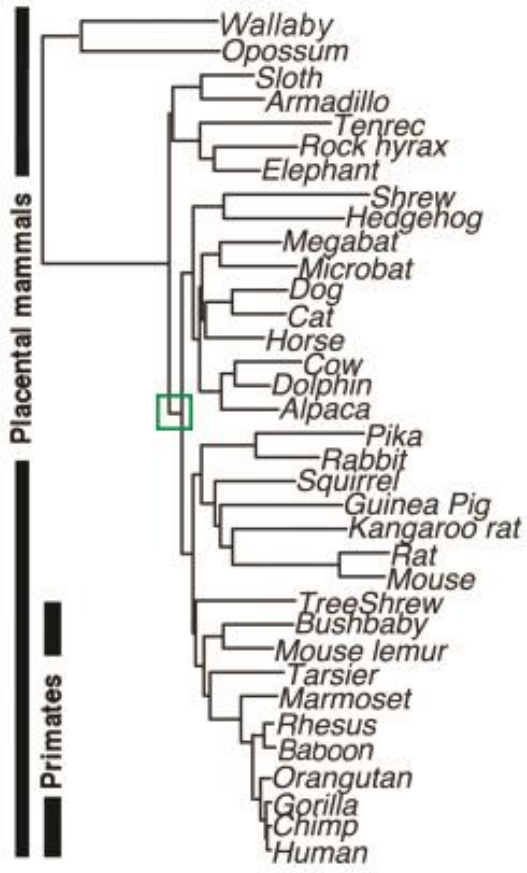
| Transcription F | MOTIF # | Status | Transcription F | MOTIF # | Status |
|---|---|---|---|---|---|
| SP1 | *M00008* | Not Used | POU2F2 | *M00795* | Used |
| NF-E2 | *M00037* | Used | USF | *M00796* | Used |
| ERalpha | *M00191* | Not Used | c-Myc | *M00799* | Not Used |
| GATA-1 | *M00347* | Used | Egr-1 | *M00807* | Used |
| GATA-2 | *M00348* | Used | 2-Oct | *M00930* | Used |
| GATA-3 | *M00350* | Used | SP1 | *M00931* | Used |
| AP-2alpha | *M00469* | Used | E2F1 | *M00940* | Used |
| AP-2gamma | *M00470* | Used | GR | *M00955* | Used |
| STAT1 | *M00496* | Not Used | ETS1 | *M00971* | Used |
| STAT3 | *M00497* | Not Used | RFX5 | *M00975* | Used |
| ERalpha | *M00511* | Used | EBF | *M00977* | Used |
| ATF3 | *M00513* | Used | TAL1 | *M00993* | Used |
| E2F | *M00516* | Used | Pbx3 | *M00998* | Used |
| Nrf1 | *M00652* | Used | SRF | *M01007* | Used |
| PU.1 | *M00658* | Used | HSF1 | *M01023* | Not Used |
| TCF4 | *M00671* | Used | NRSF | *M01028* | Not Used |
| ELF1 | *M00746* | Used | HNF4 | *M01031* | Used |
| IRF1 | *M00747* | Used | PAX5 | *M90014* | Used |
| SREBP1 | *M00749* | Used | FOXA2 | *M90047* | Used |
| CEBPB | *M00770* | Not Used | MEF2A | *M90052* | Used |
| NFKB | *M00774* | Used | GABP | *M90062* | Used |
| NF-YA | *M00775* | Used | ELK4 | *M90076* | Used |
| YY1 | *M00793* | Used | TBP | *M90108* | Used |

Reason for Not Used:

M00008: Use M00931 instead
M00101: Use M00511 instead
M00496: No Prediction File Available
M00497: No Prediction File Available
M00770: Sample Size Too Large for Computation
M00799: Binding Sites Not Available in Prediction File
M01023: Insufficient Data to Generate Predictor
M01028: Sample Size Too Small and Skewed Towards Functional

## A-2 Species

| Species | UCSC Version Number |
|---|---|
| **Human** | HG19 |
| **Chimp** | PANTRO2 |
| **Gorilla** | GORGOR1 |
| **Orangutan** | PONABE2 |
| **Rhesus** | RHEMAC2 |
| **Baboon** | PAPHAM1 |
| **Marmoset** | CALJAC1 |
| **Tarsier** | TARSYR1 |
| **Mouse Lemur** | MICMUR1 |
| **Bushbaby** | OTOGAR1 |
| **Tree Shrew** | TUPBEL1 |
| **Mouse** | MM9 |
| **Rat** | RN4 |
| **Kangaroo Rat** | DIPORD1 |
| **Guinea Pig** | CAVPOR3 |
| **Squirrel** | SPETRI1 |
| **Rabbit** | ORYCUN2 |
| **Pika** | OCHPRI2 |
| **Alpaca** | VICPAC1 |
| **Dolphin** | TURTRU1 |
| **Cow** | BOSTAU4 |
| **Horse** | EQUCAB2 |
| **Cat** | FELCAT3 |
| **Dog** | CANFAM2 |
| **Microbat** | MYOLUC1 |
| **Megabat** | PTEVAM1 |
| **Hedgehog** | ERIEUR1 |
| **Shrew** | SORARA1 |
| **Elephant** | LOXAFR3 |
| **Rock Hyrax** | PROCAP1 |
| **Tenrec** | ECHTEL1 |
| **Armadillo** | DASNOV2 |
| **Sloth** | CHOHOF1 |
| **Opossum** | MONDOM5 |
| **Wallaby** | MACEUG1 |




Lists all the species whose genome was used in our analysis. The UCSC genome version is listed with the species and the phylogenetic tree is shown above.