Extracting Regulatory Signals from DNA Sequences using Syntactic Pattern Discovery

by

Vipin Gupta

Submitted to the Department of Chemical Engineering
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Chemical Engineering

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

[September 2004]
Aug 2003

Author .....
Department of Chemical Engineering
Aug 15, 2003

Certified by .....
Gregory Stephanopoulos
Professor
Thesis Supervisor

Accepted by .....
Daniel Blankschtein
Chairman, Department Committee on Graduate Students

Extracting Regulatory Signals from DNA Sequences using Syntactic Pattern Discovery

by

Vipin Gupta

Submitted to the Department of Chemical Engineering
on Aug 15, 2003, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Chemical Engineering

Abstract

One of the major challenges facing biologists is to understand the mechanisms governing the regulation of gene expression. Completely sequenced genomes, together with the emerging DNA microarray technologies have enabled the measurement of gene expression levels in cell cultures and opened new possibilities for studying gene regulation. A fundamental sub-problem in unraveling regulatory interactions in both prokaryotes and eukaryotes is to identify common binding sites or promoters in the regulatory regions of genes. For a gene's mRNA to be expressed, a class of proteins called transcription factors must bind to the cis-regulatory elements on the DNA sequence upstream of the gene, to enhance RNA polymerase binding and hence initiate transcription. These binding sites are believed to be located within several hundred base pairs upstream of the respective ORFs.

Biological methods for discovering regulatory binding sites are slow and time consuming. To address this problem, several heuristic-based computational methods have been developed in the past with either of two approaches — *sequence-driven* or *pattern-driven*. In this dissertation, we propose a novel approach for finding shared motifs in DNA sequences based on an exhaustive pattern enumeration algorithm, that combines the benefits of the pattern-driven and sequence-driven approaches. We developed TABS, a method that identifies local regions of high similarity by clustering statistically significant patterns to obtain putative binding sites. The method assumes minimal *apriori* information about the sites and can detect signals in a subset of the input sequences, making it amenable for motif-discovery in gene clusters obtained from microarray experiments.

The performance of the algorithm was validated on synthetic as well as real datasets. When tested on a set of 30 well-studied regulons in *Escherichia Coli*, with

known instances of regulatory motifs collected from biological literature, the algorithm showed, in 14 cases, a high sensitivity and specificity of 70% and 80%, respectively. TABS was shown to perform better than two other popular state-of-the-art motif-finding algorithms. In addition, its applicability on synthetic microarray-like data was demonstrated. Several significant novel motifs detected by the algorithm that form good targets for investigation of regulatory function by biological experiments were reported.

Thesis Supervisor: Gregory Stephanopoulos
Title: Professor

Acknowledgments

I would like to express my deepest gratitude to my advisor, Professor Gregory Stephanopoulos, for providing me with appropriate guidance at each stage of my project, for his constant encouragement of my work, and for giving me the freedom to pursue research directions of interest to me. Dr. Isidore Rigoutsos with whom I worked closely during my project, is owed many thanks for his insightful comments, constructive criticism and timely guidance. I would also to thank Professor Gregory J. Mcrae, Prof. Joanne Kelleher and Prof. Chris Kaiser, the other members of my thesis committee.

Kyle Jensen is owed immeasurable thanks for being a supportive labmate all throughout this thesis work. Without his leads and enthusiastic drive for new and exciting research projects, my experience in the lab would not have been half as rewarding. Besides keeping the environment of the lab lively and jovial, he managed to amuse everybody (especially the Indian community around) with his spontanenous "context-specific" delivery of Hindi phrases!. My special thanks go to Bill Schmitt with whom I have spent innumerable hours of fruitful discussions on a wide variety of topics, from research-related material to daily news. From our seminar-discussions to gymnasium-sessions, Bill's company has always been an enriching and learning experience. I would also like to acknowledge other members of the Bioinformatics group, Daehee Hwang, Faisal Reza and Jatin Misra for their invaluable friendship and support. My thanks also go to members of Metabolic Engineering lab, particularly, Pete Heinzelman with whom I shared insighful discussions on the experimental and biological aspects of my research.

Amidst the ongoing daily stress, Susan Lanza gave us the motherly support and care, showering us with sweets and cakes, that was much needed. Also, special thanks to Brett Roth for his prompt attention to the lab needs; not to forget the several interesting anecdotes that he narrated from his vast experience in my every visit to

the office.

This acknowledgments would not be complete without a mention of some of my close friends that I was fortunate to have the company of, all through these years. Particularly, I'd like to mention Anoop V. Rao for motivating in me the strength to put in long hours while writing my thesis, Mahesh Kumar for his prompt help in statistics, and Ashish Nimgaonkar with whom I shared several late-night discussions on the "future of Bioinformatics".

Finally, infinite gratitude to my parents, my brother and my sister-in-law for providing me the emotional and moral support, that instilled in me the confidence to pursue my research-related endeavors during this eventful period of my life.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The elucidation and understanding of cellular functions has been long sought by mankind. The potential applications that this understanding bequeaths, especially in the area of medicine and drug discovery, are immense. The past few years have seen a tremendous explosion in data that can be used to describe cell physiology. With the advent of new and high-throughput genomics-based technologies that enable rapid genome-sequencing as well as probing of intracellular space, the vision of systems biology is increasingly becoming realizable.

The cell comprises various biological entities such as enzymes, metabolites, RNA, transducers, receptors, that are in a continuous state of interaction with one another and external stimuli to perform various biological functions, of which we understand only a very small fraction. The basic programming of a cell is essentially established by the genome. Although seemingly static, the cell applies this programming in an inherently dynamic manner that is dependent on the cell's interaction with its environment. Different genes get activated or inactivated to varied extents depending on dynamic environmental conditions, so at any one time, only a fraction of the cell's genetic programming is active. DNA microarray technologies enable us to measure

the relative abundance of the entire transcriptome (mRNA) which allows identification of groups of genes that are co-expressed in the cell and provides insight about interactions between various genes. By studying *gene regulatory networks* the regulatory impact of genes on other genes may be uncovered. These interactions can occur directly or through intermediate molecules.

Specific regulatory signatures in DNA sequences called *promoter* sequences encode a wealth of knowledge about regulation. These sequences recruit special proteins that bind to them and affect transcription of proximate protein-coding regions. Location of these signatures in the chromosome and knowledge about the proteins that recognize them, can provide valuable information of how and when specific genes are turned "on" or "off" in the cell, leading to the understanding of cellular control mechanisms. This thesis addresses the problem of discovering regulatory signatures in genomic sequences, *de novo*, using only sequence information.

1.1 Gene Regulation and Transcription Initiation

DNA is the carrier of genetic information in cells. It is made of deoxyribonucleotides arranged in a linear fashion as a double-helical structure (see Figure 1-1). Segments of DNA, called genes, that code for various biological activities, are expressed inside different cells and tissues in varying amounts at different stages of the cell cycle, through the process of *transcription*. The transcribed genes or mRNAs, as they are called, are further *translated* into biologically active proteins such as enzymes which take part in the various metabolic and cellular processes continuously occurring in the cell. Together, the process of transcription and translation (along with some post-translational modifications) determine the concentration levels of various biologically-active species in a cell at any point of time, and hence govern or *regulate* cellular functions. Transcription, being the first step, provides a primary control for gene

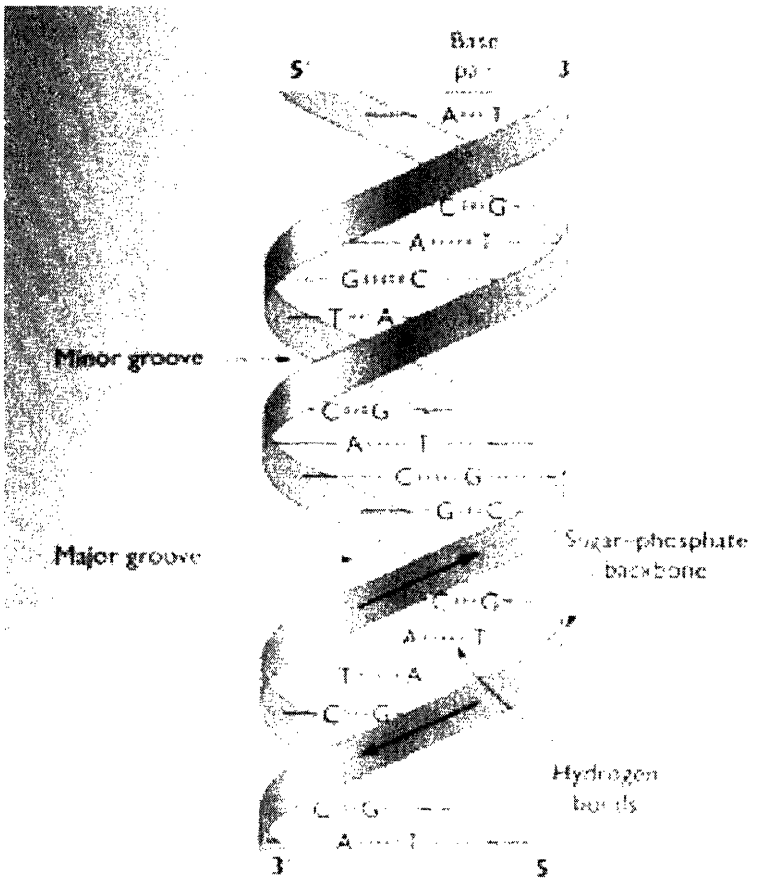


Figure 1-1: The double helix structure of DNA (figure excerpted from [5]).

regulation in this process.

1.1.1 Transcription Initiation

Transcription occurs when an enzyme called RNA polymerase reversibly binds to a certain portion on the chromosome, close the transcription start site (TSS) for a particular gene. The RNA polymerase unfolds the double-helical DNA structure and transcribes the downstream gene into an mRNA (see Figure 1-2). In certain cases, a regulatory protein comes and attaches to a site on the chromosome very close to

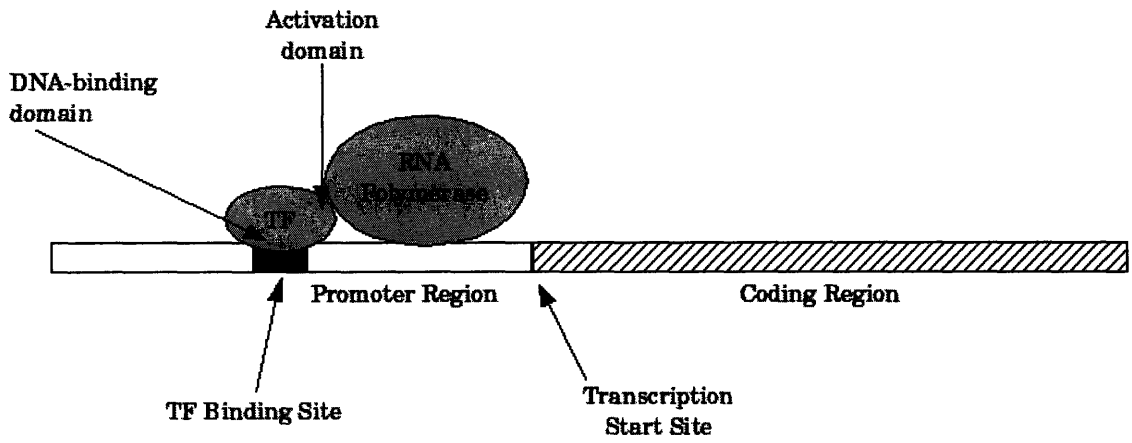


Figure 1-2: Role of DNA-Protein binding in gene-regulation during transcription initiation

the RNA polymerase, either enhancing or inhibiting the DNA-binding affinity of the polymerase. Such proteins form a distinct class called *transcription factors* (TFs). Transcriptions factors have at least two functional domains, a *DNA-binding domain* which recognizes specific DNA structure/sequence near a gene, and an *activation domain* which allows them to interact with the RNA polymerase. If the interaction inhibits the transcription process, the TF involved is called a *inhibitor* or *repressor*; if it enhances its called an *enhancer*. Region near the gene where DNA-binding takes place is called *promoter* and the specific sites where TFs bind are called *operators* or *cis*-acting sites or simply *binding sites*.

TFs, thus, act as "switches" that can turn a gene "on or off" without directly affecting the expression of other genes. The regulated gene itself may code for a transcription factor which in turn regulates another gene, forming a cascade of regulatory effects also called *gene regulatory networks* (Figure 1-3). The different arrangements of such regulatory motifs within various promoters, together with the cell-type specific

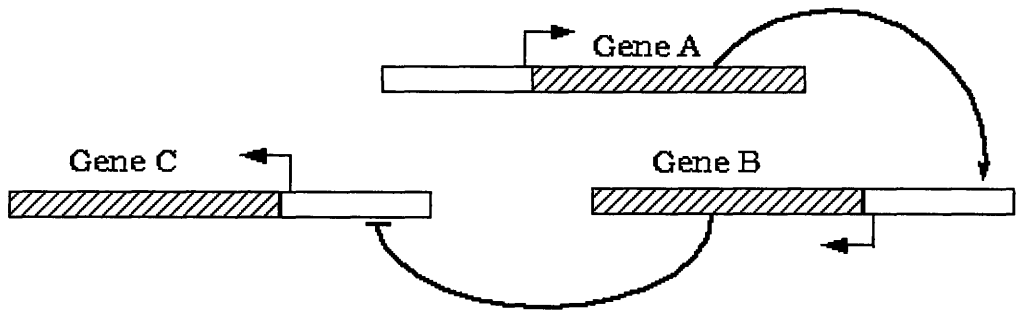


Figure 1-3: Cascade of regulatory processes forming a gene regulatory network: Gene A which is induced/suppressed by a particular transcription factor codes for a protein that induces the transcription of Gene B. In turn, Gene C is repressed by the protein coded by Gene B.

expression pattern of transcription factors interacting with them, leads to the regulation of numerous biological phenomena in eukaryotic and prokaryotic cells through complex regulatory networks.

A study of transcription regulation is crucial for understanding the cell. Whether it is the routine functions that a cell performs to grow and replicate, or the information processing and response mechanisms that are deployed by the cell to deal with external stimulus, transcriptional regulation is heavily utilized as the building block of elaborate cellular mechanisms. A detailed study of this topic is presented by Weinzierl [44].

1.1.2 Tryptophan(trp) Operon

To illustrate how regulation takes place inside cells, we present an example of tryptophan biosynthesis. Tryptophan is an amino acid required in the synthesis of peptides in cells. Among the several enzymes involved in the production of tryptophan from

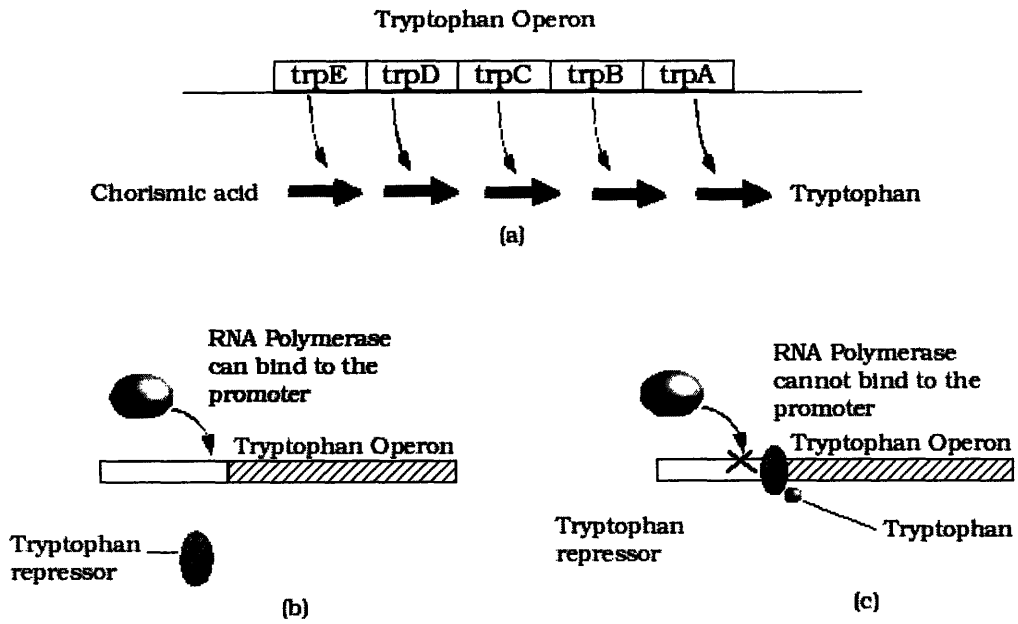


Figure 1-4: (a) The tryptophan operon consisting of 5 key enzymes involved in the synthesis of tryptophan. (b) NO TRYPTOPHAN: Tryptophan repressor cannot attach to the operator. (c) TRYPTOPHAN PRESENT: Repressor-tryptophan complex attaches to the operator

chorismic acid, five of them are encoded by genes trpA-trpE in the tryptophan operon (see Figure 1-4(a)). In the promoter region of this operon there exists a *cis*-regulatory site for a repressor called "tryptophan repressor", that when bound, prevents the transcription of genes in the tryptophan operon. However, the repressor binds to the site only in the presence of tryptophan, enabling the cell to control the production of tryptophan by "turning on" the tryptophan biosynthesis pathway *only* when there is no tryptophan present (see Figure 1-4(b),(c)).

1.1.3 Binding Site Motifs

Locating the positions of DNA-binding sites in a genome

Often the first thing that is discovered about a DNA-binding protein is not the identity of the protein itself, but the features of the DNA sequence that the protein recognizes. Various methodologies have been developed to identify the sites on the DNA where DNA-protein occurs. For instance, gel-retardation techniques make use of the substantial difference between the electrophoretic properties of a 'naked' DNA fragment that carries a bound protein, to indicate the location of a protein binding site in a DNA sequence. In this technique, a nuclear extract is mixed with a DNA restriction digest containing DNA fragments spanning the region that is suspected to contain a protein binding site, and run through a gel. The banding patterns reveal the fragments some of which get retarded in the gel. The retarded fragments contain a bound protein which leads to an increase in weight. Gel-retardation assays give a general indication of the location of protein binding site in a DNA sequence, but do not pinpoint the site with great accuracy. Often the retarded fragment is several hundred base pairs in length, compared with the expected length of a binding site of a few tens of base pairs. Retardation assays are therefore a starting point; other assays, such as *DNA footprinting* and *deletion mutagenesis*, take over where gel-retardation assays leaves off.

The basis of DNA footprinting is that if a DNA molecule carries a bound protein then part of the nucleotide sequence will be protected from "modification". The modification could be created, for instance, by treatment of DNA with a nuclease which cleaves all phosphodiester bonds except those protected by the bound protein. Figure 1-5 schematically depicts this procedure. End-labeled restriction fragments, mixed with nuclear extract, are treated with a nuclease under limiting conditions such as low temperature, so that on an average each copy of the DNA fragment is

cleaved only once. In the entire population of fragments, all bonds are cleaved except those protected by the bound protein. The protein is now removed, the mixture electrophoresed, and the labeled fragments visualized. The result is a ladder broken by a blank area which corresponds to the footprint od the binding site of the protein.

Deletion mutagenesis, another assay for localizing binding site position on the DNA, works by systematic deletion of upstream promoter regions in steps of a single base pair, and noting the subsequent changes in gene expression levels. A no change in gene expression implies that the region does not participate in regulation, an increase in the gene expression suggests the presence of an enhancer-inducing site, and similarly a decrease suggests a repressor-binding site. This methodology, thus, also provides information about the nature of the regulatory interaction.

Once identified, a binding site can be used to purify the DNA-binding protein (by methods such as affinity chromatography), as a prelude to more detailed structure studies involving X-ray crystallography techniques. Using these methodologies, literally hundreds and thousands of prokaryotic and eukaryotic promoters have been isolated and the corresponding transcription factors biochemically characterized.

These studies reveal interesting characteristics of transcription factors, DNA-protein complexes and binding sites, none of which are completely understood yet. Many TFs have at least one distinct DNA-binding domain (such as the *helix-turn-helix* domain shown in Figure 1-6), which is used to recruit the transcription factors to the promoter regions of distinct sets of genes within the genome. The sites, which they bind to, can be as short as a few base pairs or as long as 60bps. The specificity of these DNA-binding domains can vary greatly too. TFs that bind with a low degree of sequence-specificity could be used to regulate the activity of a wide range of genes, but may not be sufficiently accurate for controlling the expression of very specific subsets of genes involved in highly specialized cellular events. Further, sequence-specificity is governed by the formation of DNA-protein complexes. TFs interact with the DNA

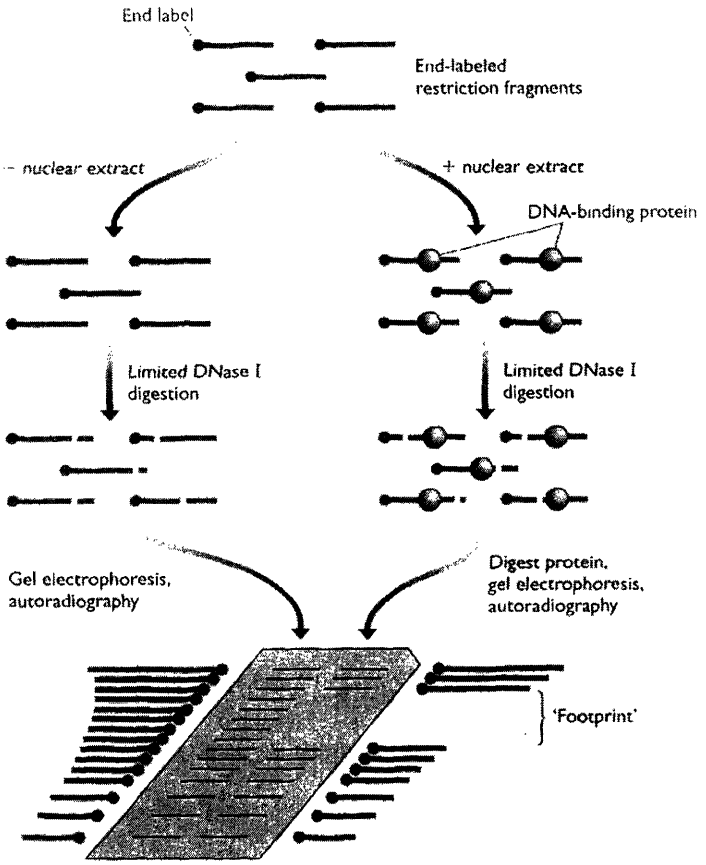


Figure 1-5: DNA footprinting: Restriction fragments (obtained from gel-retardation assay and containing promoter regions) used at the start procedure are labeled at one end. One sample of the labeled fragments is treated with a nuclear extract, while another is not. DNase I is used to cleave every phosphodiester bond, leaving only the DNA segments protected by the binding protein. The treatment is carried out special conditions so that on an average each copy of the DNA fragment is cut just once. The protein is then removed, the mixture electrophoresed. Upon visualization, one finds a ladder of bands corresponding to fragments that differ in length by one nucleotide, except in a blank area called 'footprint', corresponding to the binding site (excerpted from [5]).

Figure 1-6: One type of DNA-binding domain: helix-turn-helix loop (excerpted from [5]).

either through the major groove or the minor groove. If the interaction is through the minor groove distinction can only be made between (A/T) base pairs and (C/G) base pairs, whereas, full recognition of the nucleotide sequence is possible for proteins that contact bases through the major groove [44].

Properties of Binding Sites

Transcription factors rarely have an absolute requirement for a precise sequence motif in their target DNA. These proteins usually bind to a host of similar and interrelated sequences. Nevertheless, alignment-based comparison of individual members of such a family bound by a particular TF allows the identification of some form of consensus sequence that presents an idealized binding site. The consensus sequence is a DNA motif that the TF would predictably bind with a high affinity. For instance, the GAL4

protein in yeast recognizes all sites represented by the 17bp consensus CGGN_{11}CCG, where the 11 spacer positions in the middle could be occupied by any nucleotide base pair.

Two common ways to represent a motif are *weight matrices* and *consensus strings.* A weight matrix is a matrix W such that $W_{i,j}$ is the probability of occurrence of the i^{th} nucleotide (among A,G,C,T) at position j in the motif. A consensus string is a string over the alphabet {A,C,G,T,W,S,R,Y,K,M,B,D,H,V} that captures the composition of most occurrences of the motif (where W,S,R,Y,K,M,B,D,H and V denote combinations of pairs of nucleotides — see appendix A.2). While weight matrices are more informative in their description of binding site motifs, consensus strings provide a convenient and short method of representation, that has numerous advantages in performing fast motif-based-searches in huge genomic sequences.

Often, it is found that motifs have a dyadic structure of the kind $w_1 N_x w_2$, where w_1 and w_2 are short words made from the alphabet mentioned above, and N_x is a sequence of x (fixed) spacers (wildcards). When w_1 and w_2 are reverse complements[1] of each other, as in the GAL4 case, the motif is called a *palindrome.* Palindromes are common among binding sites motifs. It is now understood that many DNA-binding proteins are dimers that have a pair of DNA-binding domains which enable them to bind to two disjoint pieces of DNA separately.

The location of these binding sites with respect to the transcription start site (TSS) is also of significance. While most *cis*-acting sites occur upstream of the gene, often as a cluster close to the TSS, other sites have been found downstream of the TSS inside the coding regions, introns or even downstream of 3' end of the gene. This is, however, more common in eukaryotes than prokaryotes. The proximity of the various TFs interacting with the RNA polymerase, in such cases, is ensured through

[1] a reverse complement of a nucleotide sequence is the complement of the sequence written in reverse order. Complement of A is T and G is C.

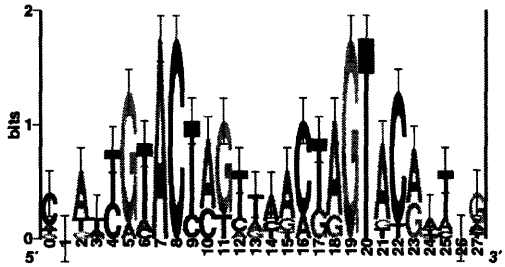


Figure 1-7: Representation of motifs using Schneider's sequence logos. Height of a nucleotide at each position is proportional to its frequency of occurrence at that position.

loop-like secondary and tertiary structures in DNA sequences.

Motifs are also known to function in a combinatorial manner. One or more genes may share the same set of motifs, cooperatively bound by different transcription factors. Such a group of motifs is referred to as a *composite element* or a *composite motif.* Composite elements are admittedly more common in promoter regions of eukaryotes than prokaryotes. For a detailed review of occurrence of composite motifs in promoter regions of genes, see Sinha [40].

Binding site motifs are often represented visually using Schneider's sequence logos which are based on the concept of Shannon's information theory [38]. In [36] and [37], Schneider explains how weight-matrices that describe the distribution of the four nucleotides A,T,G and C, at each individual position in a binding site motif, can be used to construct physically meaningful graphic logos in which each nucleotide is represented using a corresponding graphic letter with a height proportional to the information content of that nucleotide. The physical meaningfulness has to do with the fact that information content is analogous to the concept of entropy which is related to the Gibb's free energy of binding as $\delta G = -T\delta S$ (where T is temperature and S

is entropy). The concept of information content will be dealt with explicitly in section 2.1.2 later. The example of a binding site motif representation using Schneider's sequence logos is shown in Figure 1-7

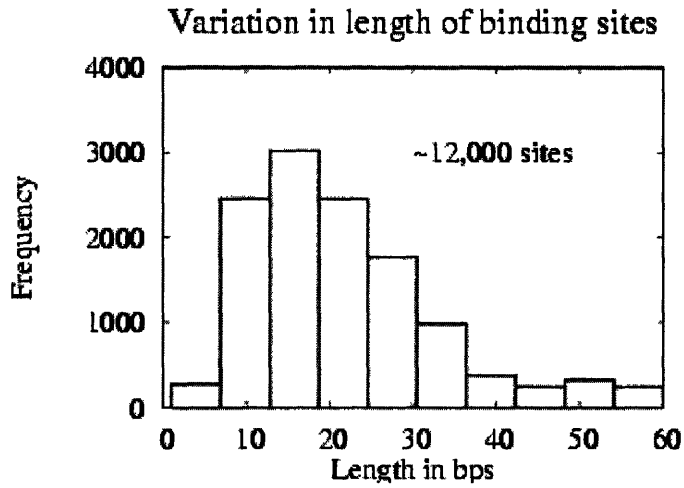


Figure 1-8: Histogram showing length of $\sim$ 12,000 sites reported in TRANSFAC database.

Databases

Several databases have sprouted in the recent past that maintain and continually update a catalogue of footprinted DNA sequences proven to demonstrate regulatory properties through painstaking experiments in wet-laboratories. TRANSFAC [45] is a database of transcription factors and their binding sites in all eukaryotes. RegulonDB [34] and DPInteract [32] are databases for *Escherichia coli* while SCPD (Saccharomyces Cerevisiae Promoter Database) [47] is a repository of regulatory information for yeast. Among these, TRANSFAC is the largest, containing records for 12,514 sites and 4,921 factors (release 6.3). A histogram showing the distribution of length of binding sites listed in TRANSFAC 6.3 is shown in Figure 1-8. RegulonDB 3.2 contains 461 E.coli sites and 86 factors.

1.2 Scope and Outline of this Thesis

The aim of this thesis is to develop methodologies for identifying binding sites in DNA sequences using syntactic pattern discovery. In Chapter 2, we begin by providing a landscape of existing computational techniques and the typical approaches that are followed to address the problem of motif-finding. In the latter part of this chapter, we describe Teiresias, an unsupervised pattern discovery algorithm developed by Rigoutsos et al. in 1998 [28], its salient features and how it can be used to develop a new motif-finding methodology that addresses the shortcomings of the existing computational methods. We then develop and describe TABS, a Teiresias-BAsed Binding Site identification algorithm that involves: (a) using Teiresias effectively to enumerate the pattern space, (b) deploying rigorous statistical tools and clustering methods to select significant pattern that target binding sites. The is done in Chapter 3. In Chapter 4 we evaluate the performance of TABS on a well-studied prokaryote, *Escherichia coli*, by basing the analysis on experimentally-proven binding sites reported in the literature for this species. We compare the results from TABS against two other state-of-the-art algorithms, AlignACE and Consensus. Finally, we conclude in Chapter 5 by pinpointing out the main contributions of this thesis, and identifying important research directions for the future.

Chapter 2

Pattern Discovery Algorithms

This chapter is divided in two parts. In the first part we begin with a description of the general approach employed for finding regulatory sites in DNA sequences and the associated challenges. This is followed by specific descriptions of current algorithmic approaches, their key characteristics and drawbacks. In the second part of this chapter, we describe a pattern discovery algorithm called Teiresias, developed by Rigoutsos, I and Floratos A, 1998 [28], where we begin by formally introducing a generic pattern discovery problem, followed by the specifics of the pattern discovery problem addressed by Teiresias, its salient features, implementation and previous applications. In particular, we address how *maximality* and *completeness*, two key characteristics of Teiresias, make it suitable for the problem of motif-finding. We conclude the chapter by motivating the development of a new motif-finding algorithm based on Teiresias that addresses issues with the existing algorithms.

2.1 Previous Algorithmic Approaches for Finding Regulatory Motifs

Typically, two modes of study are employed while finding regulatory motifs: (a) pattern-searching approach, (b) pattern-discovery approach.

Pattern-searching approach: If we know the sequences of the binding sites of a transcription factor, we can construct a template motif and use that to look for occurrences of this motif in promoter regions of other genes. Presence of the motifs is circumstantial evidence that the gene may be regulated by the transcription factor. Through wet-laboratory experiments of the kind described in section 1.1.3 the findings can be confirmed. This approach can be used to identify target sites at a genomic scale fairly easily, but is limited by the number of experimentally proven binding sites.

Pattern-discovery approach: In another type of approach, we can start with the hypothesis that a set of genes is regulated by the same transcription factor. (Such genes are said to form a "regulon"). We can then look for motifs that are shared by the promoter regions of these genes. If any such motif is found, we can experimentally verify if there exists a transcription factor that has high specificity for the motif, and if so, that transcription factor is a potential regulator of the set of genes that we started with. This kind of study is the most relevant application scenario for the algorithm presented in this work.

The *pattern-discovery approach*, used in conjunction with microarray data, forms the most commonly adopted method for discovering regulatory motifs. Microarrays provide a means for high-throughput gene expression analysis of thousands of genes simultaneously [35]. Clusters of genes found to be co-expressed can be searched for conserved words in their upstream regions. Such findings could provide key insights into the regulatory mechanisms governing the transcription of the coregulated genes

and can significantly enhance the wealth of information that can be extracted from DNA microarray experiments. There are, however, several complexities as noted in section 1.1.3, that make this problem difficult. These can be summarized as under:

1. Binding sites are typically small but highly variable in length, ranging from as low as 4bp all the way up to 60bp in some cases.

2. The degree of conservation varies significantly from position-to-position and from one motif to another.

3. Their location with respect to the transcription start site (TSS) is highly variable.

4. Depending on the pattern model that is assumed for binding sites, the space of patterns to be searched is usually very large.

5. There can be multiple binding sites present in the promoter region of the same regulon.

A number of pattern discovery methods have been proposed previously for solving the motif-finding problem. See, for example, Church et al.[33], Bailey et al.[1], Stormo et al.[14], Tompa et al.[41], Lawrence et al.[24], Collado-Vides et al.[2], etc. In all these algorithms the basic idea is the same — *to find significantly conserved words or motifs that have a low probability of random occurrence in the genome in a set of upstream nucleotide sequences.* Such a problem has no well-defined solution; different methods employ different heuristics for solving it.

These motif-finding algorithms can be broadly categorized as either *sequence-driven*, or *pattern-driven.* Algorithms in the first group model motifs as weight-matrices that assign a certain probability distribution to each residue at each position in the motif. They work by selecting sub-strings from the sequences and using them

to form alignments. The sub-strings are selected in a manner so as to maximize the "significance" of the alignment obtained. The definition of "significance" varies from algorithm to algorithm, but essentially refers to the probability of observing the alignment given a probabilistic model of the background genome. Examples of *sequence-driven* algorithms are Gibbs sampling algorithm [19], Consensus [14], MEME[1]. Characteristic features of these algorithms include: (a) assuming the length of the motif based on heuristics, (b) assuming the number of occurrences of the motif, (c) employing an iterative procedure to move from one alignment to a more significant alignment, often converging on a local optimum. A description of these algorithms has been included in the sections below.

Pattern-driven algorithms work by enumerating a predefined class of patterns[1]. The patterns are used as motif-models, and unlike *sequence-driven* methods, *pattern-driven* methods do not model probability distributions at each position in the motif, but incorporate degeneracy by employing boolean symbols. For example, W is used to represent positions in the motif that can have an A *or* a T. Such a discrete, but less informative description, allows them to search the sequence-space *exhaustively*, unlike the *sequence-driven* approaches. Example of such method are YMF [41], SPEXS [3], dyad-analysis [13], discussed in greater detail in section 2.1.4.

2.1.1 Gibbs Sampling Algorithm, Lawrence et al. [19]

Gibbs sampling is an iterative procedure, involving building motifs by repeated sampling of sub-strings from a set of input sequences, in a stochastic manner, so as to maximize the difference between the motif model and the background model (see Lawrence, et al. [19]). The method has been mathematically formulated and explained below:

[1]By patterns, we refer to strings over the alphabet {A,C,G,T,W,S,R,Y,K,M,B,D,H,V,N} (see section A.2).

Input: A set of sequence $\mathcal{S} = S^{(1)}, \ldots, S^{(n)}$ and an integer w.

Question: For each string $S^{(i)}$, find a sub-string of length at most w, so that the similarity between the n sub-strings is maximized.

Let $a^{(1)}, \ldots, a^{(n)}$ be the starting indices of the chosen sub-strings in $S^{(1)}, \ldots, S^{(n)}$, respectively. We introduce the following notations:

- Let c_{ij} be the number of occurrences of the symbol $j \in \Sigma$ among the i^{th} positions of the n sub-strings: $s^{(1)}_{a^{(1)}+i-1}, \ldots, s^{(n)}_{a^{(n)}+i-1}$.
- Let q_{ij} denote the probability of the symbol j to occur at the i^{th} position of the pattern.
- Let p_j denote the frequency of the symbol j in all sequences of $\mathcal{S}$.

Maximize the logarithmic likelihood score:

$$Score = \sum_{i=1}^{w} \sum_{j \in \Sigma} c_{ij} \cdot \log \frac{q_{ij}}{p_j}$$

To accomplish this task, the following iterative procedure is performed:

1. Initialization: Randomly choose $a^{(1)}, \ldots, a^{(n)}$.
2. Randomly choose $1 \leq z \leq n$ and calculate the c_{ij}, q_{ij} and p_j values for the strings in $\mathcal{S} \setminus S^{(z)}$.
3. Find the best substring of $S^{(z)}$ according to the model, and determine the new value of $a^{(z)}$. This is done by applying the algorithm for local alignment for $S^{(z)}$ against the profile of the current pattern.
4. Repeat steps 2 and 3 until the improvement of the score is less then ϵ.

In [33], Roth et al. applied AlignACE, an implementation of the Gibbs sampling algorithm, to three extensively studied regulatory systems in yeast: galactose response, heat shock and mating type, and found known binding sites of Gal4 transcription factor, and the cell-cycle activation motif, apart from making novel predictions.

Drawbacks:

1. Phase shift - The algorithm may converge on an offset of the best pattern.
2. The value of w is usually unknown. Choosing different values for w may significantly change the results.
3. The strings may contain more than a single common pattern.
4. The process may converge to a local maximum.

2.1.2 Consensus, Stormo et al. [14]

Consensus is based on the idea of finding alignments having maximum *information content* in a set of input sequences. Information content, which is analogous to the concept of entropy, was first used for representing alignments of binding-site motifs by Schneider [36] as eluded to before in section 1.1.3. A position specific frequency-weight matrix is constructed from a set of aligned sites and the information content I is given by:

$$I = \sum_{b,l} \frac{n_{b,l}}{N} ln \frac{(n_{b,l} + p_b)/(N+1)}{p_b} \tag{2.1}$$

where $b \in \{A, T, G, C\}$, l refers to the position in the alignment, $n_{b,l}$ is the frequency of base b at position l across the various sites in the alignment, N is the number of sequences in the alignment, and $p(b)$ is the probability of base b, based on the

composition of the genome. The term p_b appearing in the numerator of the natural logarithm term is a sample-size correction factor. Figure 2-1 illustrates the procedure for calculating I using an example.

Information content is related to differences in binding energies for different sequences, and provides a framework for computing the relative affinities of different binding sites for the same protein [42]. Conceptually speaking, information content is not very different from the log-likelihood metric used in Gibbs sampling algorithm.

Significant alignments have high information content and are rarely expected by chance. Consensus computes the significance score or p-value of a certain value of information content using a technique based on large-deviation statistics, and then minimizes the p-value using a greedy algorithm [14]. The algorithm starts by generating all possible l-mers from the input sequences and computes information content of all pairwise alignments of l-mers in the first cycle. The most significant two-sequence alignments are saved for the next cycle in which they are aligned with all l-mers not included in that alignment and the process is repeated until the alignment contains exactly one l-mer from each input sequence. This entire process is repeated for different values of l. The top p (user-specified) alignments with the best p-values are selected from the final cycle.

Drawbacks:

1. P value calculations based on large deviation statistics are known to be accurate only in certain regimes and are not valid for any type of motif. In particular this method is accurate only for finding P values of motifs with large number of sequences [26].

2. Motifs are constrained to have a fixed width by the algorithm. This is partly offset by scanning for several different widths and finding the optimal motif.

3. Each motif is expected to have exactly one instance in every sequence.

Alignment Matrix

	A	A	T	T	G	A
	A	G	G	T	C	C
	A	G	G	A	T	G
	A	G	G	C	G	T

	1	2	3	4	5	6
A	4	1	0	1	0	1
C	0	0	0	1	1	1
G	0	3	3	0	2	1
T	0	0	1	2	1	1
	A	G	G	T	G	N

Eqn 2.1 $\Longrightarrow$

Weight Matrix

	1	2	3	4	5	6	
A	1.2	0.0	-1.6	0.0	-1.6	0.0	
C	-1.6	-1.6	-1.6	0.0	0.0	0.0	
G	-1.6	0.96	0.96	-1.6	0.59	0.0	
T	-1.6	-1.6	0.0	0.59	0.0	0.0	
	1.2	0.96	0.96	0.29	0.29	0.0	3.71

Figure 2-1: Example to illustrate the calculation of information content given a set of aligned sequences. First, an alignment matrix is created showing the frequency of occurrence of each nucleotide at every position. The alignment matrix is then converted to a weight matrix using the logarithmic term in equation 2.1. The total information content for each position is listed in the row under the weight matrix and the overall information content which is obtained by summing the individual contribution from each position is shown boxed.

4. The algorithm could converge to a local minima.

5. In terms of computation time, the algorithm becomes slow for large input sizes.

Benítz-Bellón et al. [2] evaluated the performance of Consensus on 25 well studied *E.coli* regulons by measuring the ability of the algorithm to find experimentally proven binding sites. Their results showed an overall success rate of $\sim 40\%$.

2.1.3 MEME - Maximization Expectation, Bailey et al. [1]

MEME is based on a maximization expectation method that models the input sequences using a two-component finite mixture model – a motif model and a background model. It works by searching for maximum likelihood estimates of the parameters of the two models given a dataset of sequences [1]. The key advantage of

this algorithm over Gibbs sampling is that the motif found may have zero, one or more occurrences in an input sequence.

Drawbacks:

1. The algorithm may converge to a local minimum

2. Motifs are constrained to have a fixed width by the algorithm. This is partly overcome by scanning for several different widths in each run.

3. Being a two-component model, the algorithm assumes one motif per dataset, which need not be the case, since there can be multiple motifs in the same dataset.

2.1.4 Pattern-driven Approaches

Popular *pattern-driven* motif-finding algorithms include YMF by Sinha et al. [41], SPEXS by Vilo et al. [3], and Dyad-spacing by Helden et al. [13]. Each algorithm employs a different *pattern model* for finding motifs. Sinha et al. define their pattern model to be all strings over the alphabet {A,C,G,T,R,Y,S,W,N} with 0-11 N's in the center and a certain fixed number of residues. Helden et al. [13] and Li et al. [21], define all dyads of the type $w_1 N_x w_2$ (where w is a short word (3–5bps) over the alphabet {A,T,G,C}, and x could be anywhere from 0–30) as their pattern model. Vilo et al. [3] use a more unrestrictive approach to find patterns that have flexible length, and flexible wildcards. In all these cases, the enumerated set of patterns is searched in the input sequences, their frequency of occurrence is tabulated, and compared with the *expected* number of occurrences estimated by random sampling from the genome. Those patterns that are significantly over-represented in the input sequences above the background genome are considered strong candidates for being binding motifs.

Drawbacks:

1. Space of patterns considered is narrow due to computational time and space restrictions. Not all binding motifs fit the description of the pattern class used in the model.

2. Degeneracy in sites is not modeled accurately because of the discrete nature of IUPAC symbols used to construct consensus strings.

3. Enumerated patterns are not *maximal.* That is, by extending the patterns on either side one can get more specific patterns without compromising on the sensitivity.

In this thesis, we propose to develop a methodology using Teiresias, a pattern-driven search algorithm that finds all maximal patterns with variable length in a time- and memory- efficient manner. The algorithm was developed by Rigoutsos I and Floratos A [28] and is discussed in the next section.

2.2 TEIRESIAS, an Unsupervised Pattern Discovery Algorithm

In this section we begin by the description of a general pattern discovery problem, and the various types of pattern languages used to solve such problems. We then move into the specifics of Teiresias, its patterns class, problem description and implementation. The salient features of the algorithm are highlighted along with its applications in the past.

2.2.1 General Pattern Discovery Problem

The exact definition of a pattern varies from algorithm to algorithm. We define Σ [2] as the residue alphabet over which the language is formed. The set of all possible regular expressions over Σ forms a universal set of patterns, **U**. The *pattern language* or *pattern class* of a specific pattern discovery algorithm is a well-defined subset of **U**, say **C**. Every pattern $P \in \mathbf{C}$ is a regular expression that defines a language $\mathcal{L}(P)$: a string belongs to $\mathcal{L}(P)$ if it is recognized by the automaton of P. For example $\mathcal{L}$('GC.A') is the set ('GCAA','GCTA','GCGA','GCCA').

A string is said to "match" a pattern if that string contains a substring that belongs to $\mathcal{L}(P)$. The substring itself is called an *instance* of P in the original string. For example, consider S ={'AGCTA','TGCTA','AGCAA'}. The pattern 'GC.A' is present in all the three strings in S while the pattern 'GCT' is present only in strings 1 and 2. We define *support* of a pattern as the number of instances of that pattern in a given set of sequences. Therefore, support for 'GC.A' is 3, while that for 'GCT' is 2 in S. The character '.' is called a *don't care character* or a *wildcard* and indicates a position that can be occupied by an arbitrary character in Σ.

A pattern P' is said to be *more specific* than a pattern P if P' can be obtained from P by changing one or more don't care characters to residues or by appending an arbitrary string of residues and don't cares to the left of/and right of P. Thus 'AG.TA','A.CTA' are more specific than 'A..TA'.

The pattern discovery problem can be defined as follows:

Input: A set $S = \{s_1, s_2, ..s_n\}$ of sequences from Σ^* and an integer $k <= n$.

Output: All the patterns in $\mathcal{C}$ with support at least k in S.

The computational complexity of the problem depends on how general the definition

[2] In the case of DNA sequences Σ is the set of all nucleotides

of the pattern language $\mathcal{C}$ is. The simplest case is when $\mathcal{C} = \Sigma*$ in which each pattern is a string over the residue alphabet. This problem is solvable in linear time using *suffix trees* [16]. Examples of other classes are $\mathcal{C} = (\Sigma \cup \{\text{'.'}\})^*$ that allow for wildcards, or $(\Sigma \cup R)^*$, where $R = \{R_1, R_2, ..R_n\}$ is a collection of sets $R_i \subset \Sigma$, that specify patterns such as 'A{TA}G', and finally other categories that permit flexible gaps. The general pattern discovery problem stated above (along with the definitions of $\mathcal{C}$) is a hard problem as the number of patterns in the class can be exponential in the size of the input [11].

Different pattern discovery algorithms use different approaches for solving the problem. Some of them avoid a complete exploration by using heuristics and approximation techniques which work "fast", but do not necessarily find *all* patterns for a given input. Other approaches (the so called "exact" or "combinatorial" algorithms) just accept the hardness of the problem and proceed by enumerating the entire search space, which makes them inefficient in certain cases.

2.2.2 Teiresias Terminology and Problem Definition

The pattern class handled by Teiresias is $C_{Teir} = \Sigma(\Sigma \cup \{\text{'.'}\})^*\Sigma$, that is all patterns that begin with a residue and end with a residue, but can contain any number of wildcards and residues in the middle. For example 'G..CC.A' is a pattern that belongs to this class.

Problem Definition:

Input: A set $\mathcal{S} = \{s_1, s_2, .., s_n\}$ of sequences from Σ^*, and integers l, w, k where $l \leq w$ and $2 \leq k \leq n$.

Output: All maximal $< l, w >$ patterns in C_{Teir} with support at least k in the set $\mathcal{S}$, where a pattern P is a $< l, w >$ pattern iff *every* subpattern of P with length w or more, contains at least l residues.

Associated with every pattern P is an offset list $L_S(P)$ defined as:

$$L_S(P) = \{(i,j)|\text{sequence } s_i \in S, \text{ matches } P \text{ at offset } j\}$$

A pattern is said to be *maximal* with respect to S if there exists no pattern P' which is more specific than P and such that $|L_S(P)| = |L_S(P')|$. Conversely if P is not maximal then there exists a maximal pattern P' such that $|L_S(P)| = |L_S(P')|$.

2.2.3 Salient Features of Teiresias

Teiresias has the following salient features:

1. It is based on a combinatorial approach that finds the solution space without having to enumerate the entire search space. This makes the algorithm efficient.

2. Unlike existing methods, the patterns generated are *maximal*. That is, it is not possible to make them more *specific* without reducing their *support*. This ensures that the patterns found are highly specific and there is no redundancy in them.

3. Experimental results suggest that the algorithm is *output sensitive*, i.e., its running time is quasi-linear to the size of produced output [28].

4. Unlike most existing methods it can efficiently handle patterns of *arbitrary* length.

2.2.4 Implementation

Teiresias works in two phases - *scanning phase* and *convolution phase*. During the scanning phase it enumerates all *elementary* patterns that have exactly l residues

and meet the $< l, w >$ requirement. The algorithm works by constructing $< l, w >$ template, which is defined to be an arbitrary string of 0's and 1's that has a length between l and w, contains exactly l 1's, and starts and ends with a 1. By sliding this template along the input sequences, one can construct elementary patterns by maintaining all residues that are aligned with a '1' and converting those that align with a '0' into wildcards. The time complexity of this phase of the algorithm is $\mathcal{O}(mw^l)$, where m is the number of characters in the input sequences. For further details about the working of the algorithm and formal proofs, see [11].

In the convolution phase of the algorithm, these elementary patterns are pieced together (in a time and space efficient manner) to make them maximal. For example, consider two elementary patterns 'AGC.T' and 'C.TGG'. It is possible to *convolve* these patterns into one single pattern 'AGC.TGG' based on the similarity of their suffix and prefix. In doing so, if the offset list of the new pattern (which can be constructed from the offset list of the two elementary patterns) has a size $>= k$, convolved pattern is accepted else it is rejected. Instead of resorting to an-against-all approach that would mean examining all pairs of patterns to establish if they are convolvable or not, Teiresias uses a special algorithm to quickly identify and discard patterns that not maximal in a time and space efficient manner and still generate *all* the patterns. The procedure involves two partial orderings on the list of patterns while convolving patterns. See [28] for more details. The overall time complexity of the algorithm has been experimentally found to be quasi-linear with respect to the output size (number of patterns produced).

2.2.5 Applications

In [28], Rigoutsos et al., demonstrate the strength of their algorithm by finding highly specific patterns corresponding to two protein families - histones (H3 and H4 families),

and laeghemoglobins. Some of the discovered patterns were found to match domains in the PRODOM, a curated database of protein domains [23].

Teiresias has been used for building a Bio-Dictionary which is based on the idea that all biological sequences contain small domains of functionally active seqlets [29]. The concept of bio-dictionary has been used successfully for prokaryotic gene-finding and for protein annotation. A web-server [17] hosted by IBM provides a wide variety of computational biology tools built using Teiresias — association discovery, protein annotation, gene finding, discovery of tandem repeats, multiple sequence alignment, genome annotation, gene expression analysis, etc. A detailed discussion of these applications is beyond the scope of this dissertation and the interested reader is referred to [31], [39], [30] and [17].

To reiterate, the promise that Teiresias holds against all conventional methods is (a) maximality (b) completeness. In the next section we discuss the motivation for using Teiresias for the problem of identification of regulatory motifs in DNA sequences discussed in chapter 1.

2.3 Motivation for using Teiresias for Motif-finding

In this section we propose how Teiresias could be used effectively for developing a new approach that offsets the shortcomings of the previous approaches. As discussed in section 2.1 current motif-finding algorithms belong to either of two categories — sequence-driven, or pattern-driven. Sequence-driven algorithms use weight matrices to represent motifs that provide an accurate description of degeneracy at each position in the motif. Due to the number of possible combinatorial alignments that can be generated from a given set of sequences, these algorithms employ heuristics to approximate the solution and typically don't end up scanning the entire solution space defined by the sequences. Pattern-driven approaches, on the other hand, use a

consensus-string based model that, although more restrictive in its description of binding sites, allows exhaustive scanning of the solution space defined by the sequences. However, computational time and space complexities involved in enumeration and search of patterns forces the models to be constrained in terms of length and the number of wildcards.

We propose an alternate approach using Teiresias that can address these issues. This approach promises the following:

1. Ability to scan "entire" pattern space defined by the sequences (shortcoming of sequence-driven approach).

2. Ability to find all *maximal* patterns of *arbitrary* length in a time- and space-efficient manner. (shortcoming of pattern-driven and sequence-driven approach)

3. Ability to account for degeneracy in motifs which can otherwise not be modeled effectively by pattern-driven approach. (shortcoming of pattern-driven approach)

While points (1) and (2) are a direct fallout of using Teiresias, we illustrate (3) using an example shown in Figure 2-2. The figure shows 14 binding sites corresponding to the TyrR transcription factor that have been previously reported in literature. The sequence logo obtained by performing a multiple alignment of the 14 sequences is also shown in the figure. We do a simple pattern discovery test to find what kind of patterns best describe this set. For $l > 3$ no pattern with a full support ($k = 14$) is found, i.e. there is no pattern with at least 3 residues that is matched by all the sequences. By fixing $l = 6, w = 20$ we systematically drop k to allow patterns with a smaller support. Until $k = 11$ no patterns are found. At $k = 10$ we find 2 patterns, which describe 10 of the 14 sequences. At $k = 8$ we have a total of 18 patterns that describe all the sequences, among which 3 patterns are sufficient to cover all the

sequences. These 3 patterns include the two patterns with $k = 10$ and one pattern with $k = 8$. The figure shows what these patterns look like and maps the location of the instances of these patterns on the sequences. If we had a way to group these patterns into one motif, we could obtain the original set of sequences. This idea forms the motivation for the development of a methodology which is presented in chapter 3.

```
Seq No 1-------10 11-------21 22-------32 33------42
Seq 01 caaacttctt TGATGTAAACA AATTAATACAA caaacggaat
Seq 02 ttaatacaac AAACGGAATTG CAAACTTACAC acgcatcact
Seq 03 agaaccatcg CGTGTTTCAAA AAGTTGACGCC tacgctggcg
Seq 04 tttacaccat ATGTAACGTCG GTTTGACGAAG cagccgttat
Seq 05 tgctttttat TGTACATTTAT ATTTACACCAT atgtaacgtc
Seq 06 agcgaacaca ATCTGTAAAAT AATATATACAG ccccgatttt
Seq 07 tccgtctttg TGTCAATGATT GTTGACAGAAA ccttcctgct
Seq 08 tttcaaaggg AGTGTAAATTT ATCTATACAGA ggtaagggtt
Seq 09 ctaaattgcc TGTGTAAATAA AAATGTACGAA atatggattg
Seq 10 aatgtacgaa ATATGGATTGA AAACTTTACTT tatgtgttat
Seq 11 tccgttcata GTGTAAAACCC CGTTTACACAT tctgacggaa
Seq 12 gtggctaaat GTAATTTATTA TTTACACTTCA ttcttgaata
Seq 13 aaggggtgta TTGAGATTTTC ACTTTAAGTGG aatttttct
Seq 14 tcactttaag TGGAATTTTTT CTTTACAATCG aaattgtact
```

⇓ Pattern Discovery

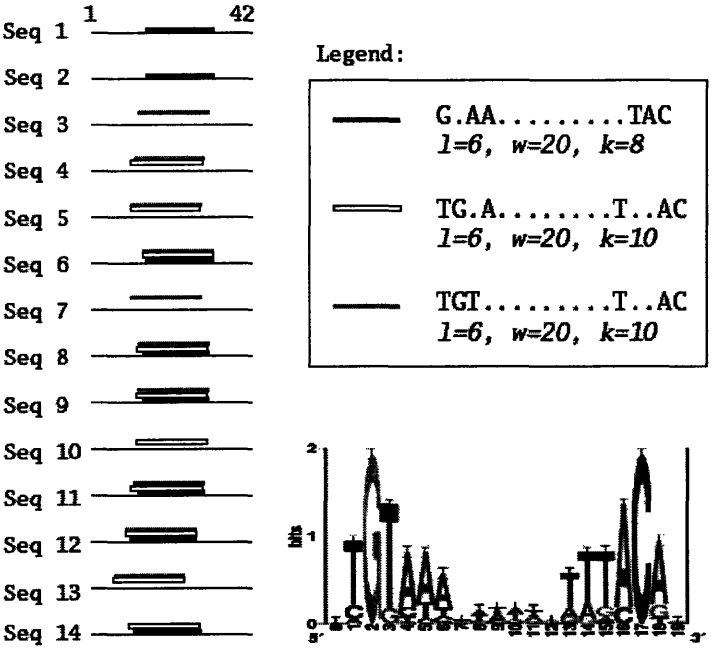


Figure 2-2: Degeneracy in TyrR binding sites: Shown on top are sequences of 14 TyrR binding sites taken from the RegulonDB database. Each site is reported as 42bps long (22bp core + 10bp flanking regions on either side). Teiresias was applied to find patterns shared between these sites. No pattern with a full support (k=14) was found for $l > 3$. For $l = 6, w = 20$, all the sequences were described using three patterns shown above, two with $k = 10$, and one with $k = 8$. The locations of the instances of these three patterns on the 14 sequences are shown above using a legend. This simple test shows the limitations of pattern-driven motif-finding approaches and presents the need to represent motifs using *more than one* patterns. Shown along with is a sequence logo (obtained independently) emphasizing the level of degeneracy in the alignment.

Chapter 3

TABS - An Algorithm for Discovering Binding Sites

In section 2.3 we motivated the development of TABS, a Teiresias-bAsed algorithm for finding Binding Sites in DNA sequences. In this chapter we present a detailed description of this algorithm. To begin with, we outline an overview of the algorithm in section 3.1. Given a set of input sequences the algorithm begins by generating an exhaustive set of patterns shared between these sequences. The appropriate choice of parameters for this step is discussed in section 3.3. Among the patterns found, statistically significant patterns that are most overrepresented in the input sequences with respect to the background genome, are selected. This is discussed in section 3.2. Finally, creation of motifs from these patterns using a novel mapping and clustering technique is described in section 3.4.

3.1 Algorithm Overview

Given a set of n input sequences TABS works in three phases as outlined in Figure 3-1.

Step 1: Basic Pattern Enumeration Step: Teiresias is used to enumerate all $l = 6, w = 20, k = k(n)$ patterns. This generates an exhaustive set of maximal patterns that have at least 6 residues or bases and a minimum density ($= l/w$) of 6/20. The relationship between k and n and the choice of l, w is discussed later in section 3.3.

Step 2: Selection of patterns based on statistical significance: A z-score statistical metric is used to estimate the degree of overrepresentation of a pattern based on its frequency of occurrence in the input sequences and the expected number of occurrences computed using a third order Markov model.

Step 3: Convolution phase: Patterns obtained from step 2 are convolved together into motifs by first mapping them on the input sequences, and then clustering them.

3.2 Evaluating Statistical Significance of Patterns

The problem of identifying significantly overrepresented words in biological sequences, or, for that matter, any general text, is both a well-studied and a computationally-intensive statistical problem. See [10] for a review of some commonly used statistical methods such as the *z-score statistic* used in DNA sequences by Sinha [41], *binomial statistic* used by Helden et al. [12] for identifying overrepresented oligonucleotides, and *large-deviation methods* for estimating p-value of alignments by Stormo [14].

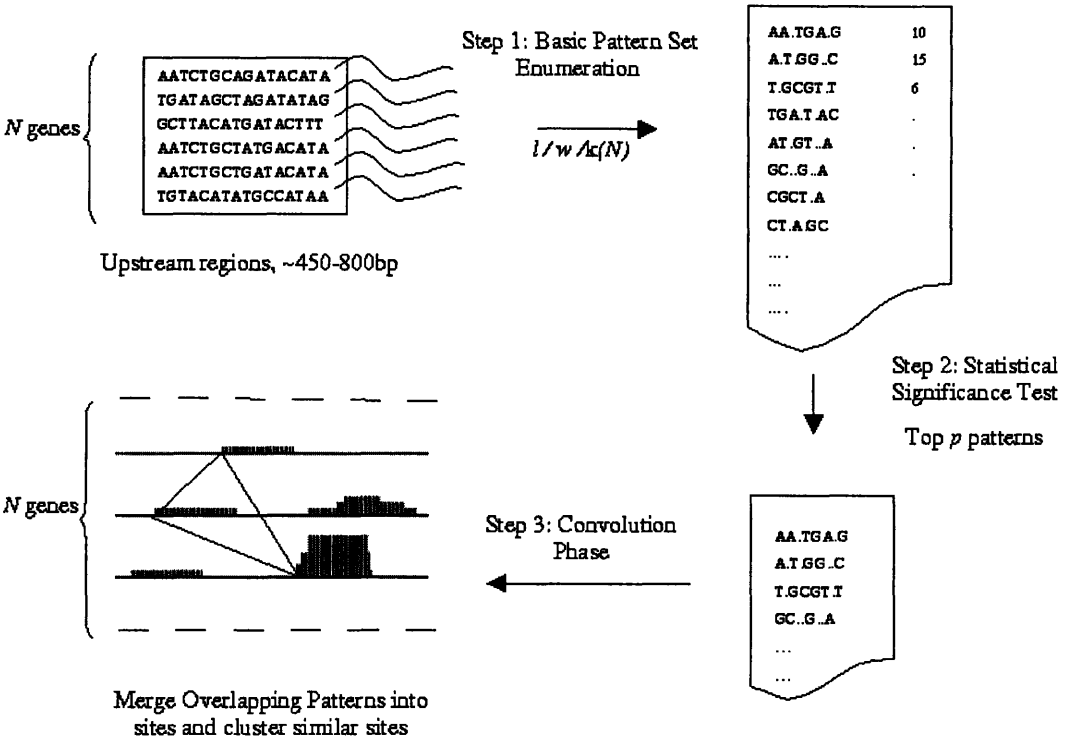


Figure 3-1: Algorithm Overview

All these methods involve computing the expected number and variance of the number of occurrences of a word, which is complicated by the problem of autocorrelation (phenomenon of overlapping occurrences of words) [18]. While this problem of autocorrelation has been explicitly dealt with by Sinha et al. [41], in other cases it has been ignored [12], [13]. Autocorrelation does not affect the expected number of occurrences but increases the variance in certain cases [18]. The effect is significant in words that are *short* and have a periodic nature, such as AAAAA, ATATA, etc. In our case, the median length of patterns found is around 17 and the probability

of occurrence of the patterns is typically very low, less than 10^{-4}, implying that a pattern is observed once every 10,000 base pairs. Overlapping is not expected for such long and rare patterns. Hence we neglect autocorrelation effects which simplifies the calculation of variance tremendously. However, in our experiments we do check if any significant patterns found have a periodic nature and manually filter them out. Frequently, such patterns get filtered out due to their high expected number of occurrences, anyway. We employ the z-score statistic with this assumption.

Let us say there is a universal set $\mathbf{U}$ of N sequences representing the non-coding / upstream regions in the entire genome and let the set of n input query sequences be $\mathbf{S} \subset \mathbf{U}$. We define $|\mathbf{S}|$ as the sum total of the length of all the sequences in $\mathbf{S}$ in base pairs. Let $\mathbf{P}$ be the set of patterns whose statistical significance we wish to estimate and let $p \in \mathbf{P}$. The length of pattern p is represented by $|p|$ and let the number of *non-overlapping* occurrences of p in $\mathbf{S}$ be $\mathcal{O}$. Let $\mathbf{R}$ be any set of n random sequences such that $\mathbf{R} \subset \mathbf{U}$ and $|\mathbf{S}| = |\mathbf{R}|$. Finally, let X_R be the number of non-overlapping occurrences of p in $\mathbf{R}$. Then the z-score associated with observing $\mathcal{O}$ occurrences of p in $\mathbf{S}$ is:

$$z = \frac{\mathcal{O} - E(X_R)}{\sigma(X_R)} \tag{3.1}$$

where $E(X_R)$ is the expected number of occurrences of p in the random set $\mathbf{R}$ and $\sigma(X_R)$ is the associated standard deviation. The measure z is the number of standard deviations by which the observed value $\mathcal{O}$ exceeds its expectation, and is sometimes called the "normal deviate" or "deviation in standard units". For a detailed discussion about this statistic see Leung *et al.* [20]. We present the calculations for computing $E(X_R)$ and $\sigma(X_R)$ below.

p can occur in $T = |\mathbf{S}| - n * (|p| - 1)$ positions in $\mathbf{R}$. Let X_i denote the occurrence

of p at position i, $i \in 1, 2 \ldots T$:

$$X_i = \begin{cases} 1 & \text{if } p \text{ occurs at position } i, \\ 0 & \text{otherwise.} \end{cases} \tag{3.2}$$

Thus we can write:

$$\begin{aligned} X_R &= X_1 + X_2 + \ldots + X_T & (3.3) \\ E(X_R) &= E(X_1 + X_2 + \ldots + X_T) = E(X_1) + E(X_2) + \ldots + E(X_T) \\ &= T \times q & (3.4) \\ \sigma(X_R) &= \sigma(X_1 + X_2 + \ldots + X_T) \sim \sigma(X_1) + \sigma(X_2) + \ldots + \sigma(X_T) \\ &= T \times q(1-q) & (3.5) \end{aligned}$$

where q is the probability of observing p at any position in a random sequence and the approximation in equation 3.5, is that we assume X_i's are independent (which follows from ignoring auto-correlation effects). q is estimated by treating the set of sequences in $\mathbf{U}$ as a markov chain and building a Markov model from these sequences as explained in next section. Thus z-score can be rewritten as:

$$z = \frac{\mathcal{O} - T \times q}{\sqrt{Tq(1-q)}} \tag{3.6}$$

where $T = |\mathbf{S}| - n * (|p| - 1)$.

3.2.1 Markov Model

Markov chains are frequently used to model biological sequences. Briefly, a Markov chain of order n is a sequence of $X_1, X_2, X_3 \ldots$ of random variables that can take discrete values from a set $\mathbf{A} = \{a_1, a_2 \ldots a_p\}$. The sequence X_i satisfies the following

Markovian property:

$$P(X_i|X_{i-1}, X_{i-2}\ldots X_1) = P(X_i|X_{i-1}, X(i-2)\ldots X_{i-n}) \quad \forall i, \tag{3.7}$$

i.e. the conditional probability of observing X_i at any state depends on the previous n states only. A Markov Model (MM) is defined by a set of *prior* probabilities, i.e. the probability of observing a state given no prior information about the preceding states, and a set of *transition probabilities* that determine the probability of the next state given the previous n states. The basic theory of MMs is described in [15] and can be found in many other texts.

Given any sequence $S_1, S_2, \ldots S_m$ one can find the probability that the sequence was generated from a given MM by using the Bayes formula:

$$\begin{aligned} P(S_1, S_2, \ldots S_m) &= P(S_1, S_2 \ldots S_n) \times P(S_{n+1}|S_1 \ldots S_n) \times \\ &\quad P(S_{n+2}|S_2 \ldots S_{n+1}) \times \ldots \times P(S_m|S_{m-n} \ldots S_{m-1}) \end{aligned} \tag{3.8}$$

The first term in the RHS of equation 3.8 is the *prior* while the remaining conditional probability terms are the *transition probabilities.*

In the case of DNA sequences, X_i can take on values from a set of 15 possible characters obtained, **A**={A,T,G,C,W,S,R,Y,K,M,B,D,H,V,N} (IUPAC nomenclature for various combinations of nucleotides, see appendix A.2). In the past, Markov Models and Hidden Markov Models ([15]) have been used several times for finding structures in genomic sequences, for instance predicting genes [22, 6], exon-intron splicing sites [46], binding sites [41], etc.

We used a third order Markov model to build a background model by training it on all the upstream sequences in **U**. We chose n=3 in order for the background model to account for the TATA, AAAA and TTTT sequences that are ubiquitous

throughout the genome's promoter regions [41]. The *prior* probabilities and *transition probabilities* can be computed by tabulating the number of occurrences of all triplets and 4-tuples as shown in equations 3.9 and 3.10 respectively:

$$p_{prior}(a_1a_2a_3) = \frac{\mathcal{O}(a_1a_2a_3)}{|\mathbf{U}|} \tag{3.9}$$

$$p_{trans}(a|a_1a_2a_3) = \frac{P(a_1a_2a_3a)}{P(a_1a_2a_3)} = \frac{\mathcal{O}(a_1a_2a_3a)}{\mathcal{O}(a_1a_2a_3)} \tag{3.10}$$

where $a_i \in \mathbf{A}$, $|\mathbf{U}|$ is the sum total of the size of all upstream sequences in bps, and Bayes rule was used in equation 3.10. In writing these equations it was assumed that prior probabilities are independent of the position in the sequence, which has been discussed and justified by Kleffe and Borodovsky [18]. Using equation 3.8, together with equations 3.9 and 3.10, one can estimate the probability of a pattern $p \in \mathbf{P}$. For instance, for p=AATGWC (W = A or T):

$$\begin{aligned} P(AATGWC) = {} & p_{prior}(AAT) \times p_{trans}(G|AAT) \times (p_{trans}(W|ATG) + \\ & p_{trans}(C|TGW)) \end{aligned} \tag{3.11}$$

Likewise one could find the probability $P(p)$ for any arbitrary pattern p. We can now substitute $q = P(p)$ from the above equation into equation 3.6 to get the z-score of p.

3.3 Choice of Teiresias Parameters

As discussed in section 2.2, three parameters l, w, k specify the pattern language that determines the set of output patterns from Teiresias. l specifies the minimum number of residues (A,T,G or C) in the pattern, l/w specifies the minimum density (number-of-residues-to-width ratio) and k specifies the minimum support (number of *unique* occurrences in input sequences).

```
TGAATTAATATGCA
TGAGTGAATATTCT
TGAATAATCATCCA
TGAATTTTAATTCA
TGCATAAAAATTCA
TGTTGTATCAACCA
TGAATTTTAATTCA
TGCAGTATTTATGA
TGAATAAAAATACA
TGTATTTTTATTCA
TGCATGAATATTGA
TGAATAATTACACA
TGNNNNNNNNNNN
```

Figure 3-2: Alignment of 12 ArgR binding sites along with their consensus pattern

Choice of these parameters is both a tricky and key issue as it determines the basic set of patterns in the algorithm. If the parameters are too stringent one can miss key patterns, while if they are too loose, unnecessary patterns can be generated, reducing the specificity of the method. The problem becomes more complicated because binding sites have degenerate base pairs at most positions in the alignment making it difficult to represent them effectively in the Teiresias pattern language. Consider the alignment shown in Figure 3-2 of 12 binding sites from ArgR regulon in *E.coli*. The pattern that describes this alignment in the Teiresias pattern language is shown in the last row (N's represent wildcards - positions where there is more than one type of base pair present). Certainly this pattern is highly degenerate and not very interesting. However, figure 3-3 reveals the correct picture. The figure shows the position-specific-weight-alignment-matrix (PSWM) and the sequence logo for the 12 sites. There are at least five more positions in the alignment that show a high degree of conservation. If a 100% value of k is specified, Teiresias would identify just the pattern shown in Figure 3-2 which being information-poor would match numerous false positives. To approach this problem the value of k is set much lower than n (= the number of sequences in the input set) to avoid the chance of missing sites

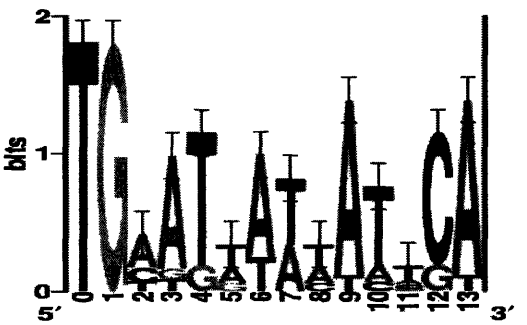
of the nature described above. This leads to generation of several similar patterns, each covering a subset of the sites, but when taken together they describe the entire set of sites (see section 2.3 for an example). The optimal value of k is case-specific depending not only on the number of input sequences but also on their structure. However, since all $k = k_1$ patterns are a subset of $k = k_2$ patterns when $k_1 > k_2$, k can be set on the lower side and the infrequent patterns can be removed in the downstream filtering step. A list of recommended k values for n from 3 to 100 is tabulated in section A.1.1. For instance for n=3, k is also set to 3, while for n=25, k is set at 9. It must be mentioned that these k values are based on trial runs on random sequences and while they have been designed to work in most cases, they may need to be changed slightly in special cases (depending on the composition of input sequences). Nevertheless, the values listed do a reasonable job in most cases as is demonstrated later in the results section.

3.3.1 Selection of l, w

Based on the length and width distributions of experimentally identified binding sites there are several feasible values of l and w, however we are interested in the optimal values that are most sensitive to binding sites. There are several factors that can aid us in deciding the best choice for l and w. The choice of w depends on l since together they dictate the *density* of a pattern. While settings such as $l = 3, w = 20$ return too many patterns with a lot of degeneracy which makes them non-interesting, $l = 6, w = 6$ finds too few patterns that are not sensitive enough. Based on such preliminary criteria and several trial runs on various different sets of sequences, eight sets of l, w were considered as possible candidates - 3/6, 4/8, 4/10, 5/15, 5/17, 6/17, 6/20 and 7/20. In order to find the best pair of l and w, a sensitivity test was performed to determine which combination produced patterns

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
A	0	0	**7**	**10**	0	4	**9**	5	4	**11**	2	2	0	**11**
T	**12**	0	2	1	**10**	6	3	7	6	1	**9**	7	0	1
G	0	**12**	0	1	2	2	0	0	0	0	0	1	2	0
C	0	0	3	0	0	0	0	0	2	0	1	2	**10**	0

PSWM for ArgR Binding Sites



Sequence Logo

Figure 3-3: Example to illustrate degeneracy in binding sites and motivate the choice of k. While only 2 positions in the logo show 100% conservation there are at least 5 more positions where one of the nucleotides is significantly conserved.

that were most sensitive known binding sites in *E.coli.*

Let **S** be the set of n input sequences, **K** be the set of *known sites* or *true sites*, and **Q** represent the set of patterns generated by Teiresias sorted on the basis of their *degree of over-representation* or *z-scores*. Further let **P** represent the set of top p patterns from the set **Q**. Finally, let $\mathbf{P} \cap \mathbf{S}$ denote the set of positions in **S** that are touched by at least one pattern in **P**. If we map the known sites onto **S**, the fraction of the positions corresponding to the known sites that are touched by **P** is given by $|\mathbf{P} \cap \mathbf{K}|$. We define two quantities:

$$Coverage = \frac{|\mathbf{P} \cap \mathbf{S}|}{|\mathbf{S}|} \tag{3.12}$$

$$Sensitivity = \frac{|\mathbf{P} \cap \mathbf{K}|}{|\mathbf{K}|} \tag{3.13}$$

The difference between coverage and sensitivity reflects the selectivity of **P** towards known sites. If sensitivity is more than the coverage, the pattern selection criteria is sensitive towards binding sites. Figure 3-4 shows a typical plot of coverage and sensitivity as a function of $|\mathbf{P}|$, the number of patterns in **P**. When $|\mathbf{P}|$ is sufficiently large, there are enough patterns to cover the all the positions in the sequences, hence both coverage and sensitivity are close to 1. As the number of patterns chosen reduces the coverage falls, ultimately becoming zero when there are no patterns in **P**.

By fixing the coverage at 20%, we compared the sensitivity of patterns generated using different l, w pairs (listed in Table 3.1 towards known sites. The dataset used for this analysis is a set of 30 known regulons described in 4.1. Of the eight combinations, $l/w = 6/20$ yielded the highest sensitivity of 43%.

A coverage of 20% was considered an optimal cut-off for deciding the number of top patterns to be retained. At this coverage the islands produced were of a reasonable size between 15-40bps (resembling the size range of known binding sites).

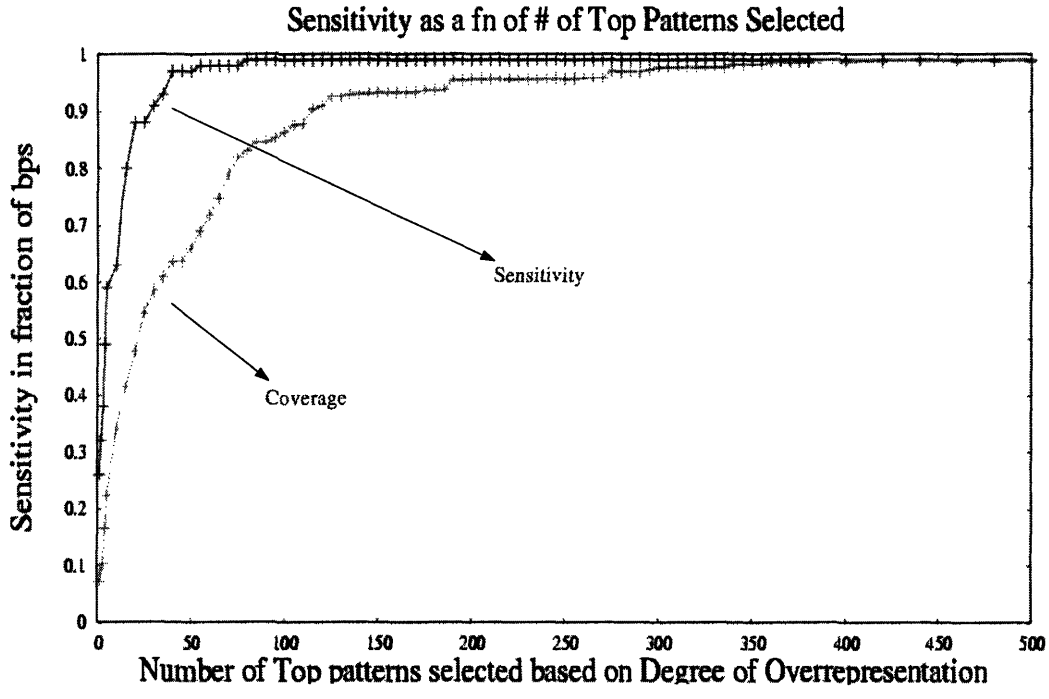


Figure 3-4: Sensitivity as a function of number of statistically significant patterns

Typically, a coverage of 20% amounts to selection of 5-20 patterns depending on the relative values of k and n. The actual calculation, showing the relationship between the number of patterns selected and the coverage, is included in the appendix section A.1.2.

3.4 Convolution Phase

The convolution phase is sub-divided into three parts: (a) building a feature map by superimposition of significant patterns on the input sequences, (b) clustering groups of similar regions in the feature map into *motifs*, and (c) creating alignments of the

l	w	Sensitivity(%)
3	6	35
4	8	28
4	10	38
5	15	37
5	17	34
6	17	35
6	**20**	**43**
7	20	31

Table 3.1: Mean sensitivity over 30 *E.coli* regulons for different pairs of l and w

motifs and ranking the motifs on the basis of the significance of their alignment. Superimposition of overlapping patterns creates a feature map which delineates *islands* of significantly overrepresented regions in the sequence space, as described in section 3.4.1. The islands are grouped into clusters as explained in section 3.4.2 to obtain the final set of *motifs*.

3.4.1 Mapping

In order to build a feature map, "sufficiently overlapping" instances of patterns are merged together. Figure 3-5 shows a portion of sequence with overlapping occurrences of several different patterns. A mapping signal proportional to the number of patterns that occur at each position in the sequence is constructed as shown. Contiguous regions on the sequences having mapping signal greater than zero are grouped together as *islands*. A 40% overlap between two successive instances of patterns is considered "sufficient". Figure 3-6 shows an example of a feature map. Henceforth, the terms "islands" and "sites" will be interchangeably used.

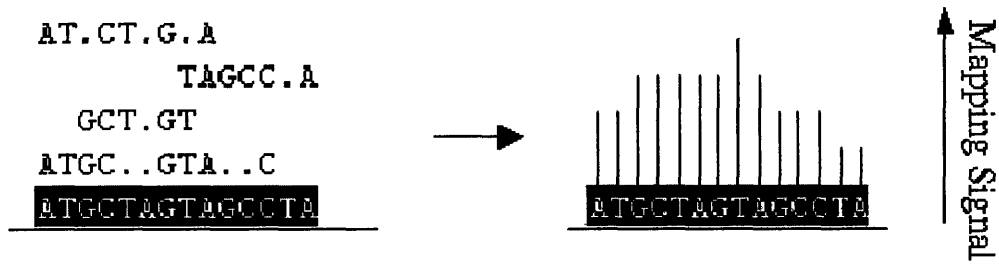


Figure 3-5: Merging of overlapping instances of patterns. The height of mapping signal at any position is proportional to the number of overlapping patterns at that position.

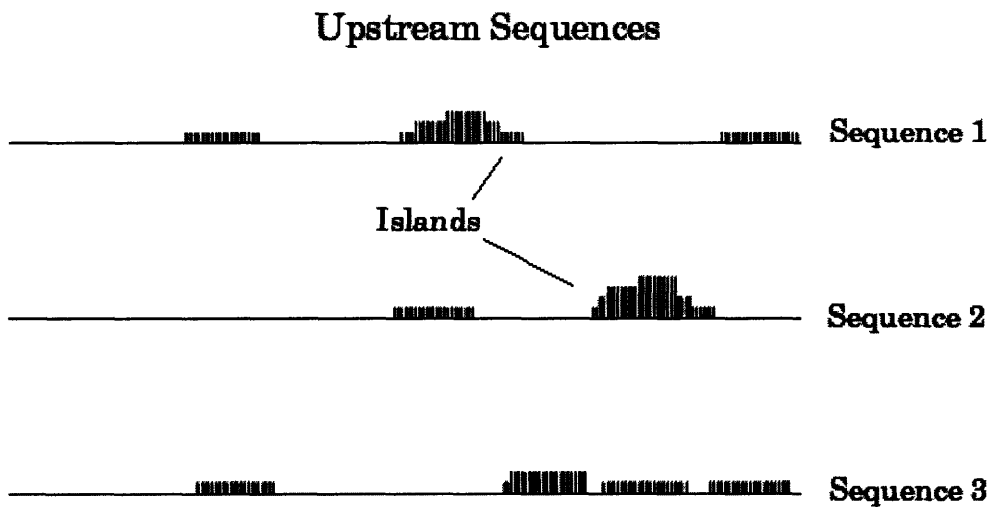


Figure 3-6: Feature map obtained by mapping significant patterns on the input sequences

3.4.2 Clustering of Sites using Graph Theory

There are several techniques available for clustering sequences, such as hierarchical clustering, k-means, etc. However, these are more suitable when the expected number of clusters is known beforehand. For the current problem, one does not know the number of motifs in a given set of sequences beforehand, hence it is difficult to estimate the number of clusters. So instead we use a graph-theory based approach which, although computationally more expensive, is more appropriate for the size of the problem at hand.

We begin by constructing a graph in which each *node* corresponds to an *island* in the feature map (Figure 3-7). All pairwise similarity scores between two nodes are found using Smith-Waterman global sequence alignment algorithm of the corresponding sites. A standard scoring matrix with a +1 score for match, 0 for mismatch and $-\infty$ for gaps (i.e.,no gaps allowed) is employed. The top $c \times n(n-1)/2$ pairs of nodes based on their similarity scores, are connected by an edge (n is the number of input sequences). c is an empirical proportionality constant which has a value of 2 or 4 (see section A.1.3 for details). Within the graph so obtained one can search for clusters of very similar islands using standard graph theory-based algorithms. Fully connected sub-graphs, also called *cliques*, are used to find tight clusters of similar sites. In a clique, each and every pair of nodes is connected by an edge (see figure 3-7). Finding maximal cliques in a graph is an NP-complete problem [8]. However several approximate algorithms exist. We used an implementation of the Bron-Kerbosch algorithm [4] for finding cliques. Among the motifs found, those having instances in a pre-specified minimum percentage of sequences are kept, while the rest are rejected. By default this cut-off is set at 70% of the input sequences but can be modified by the user.

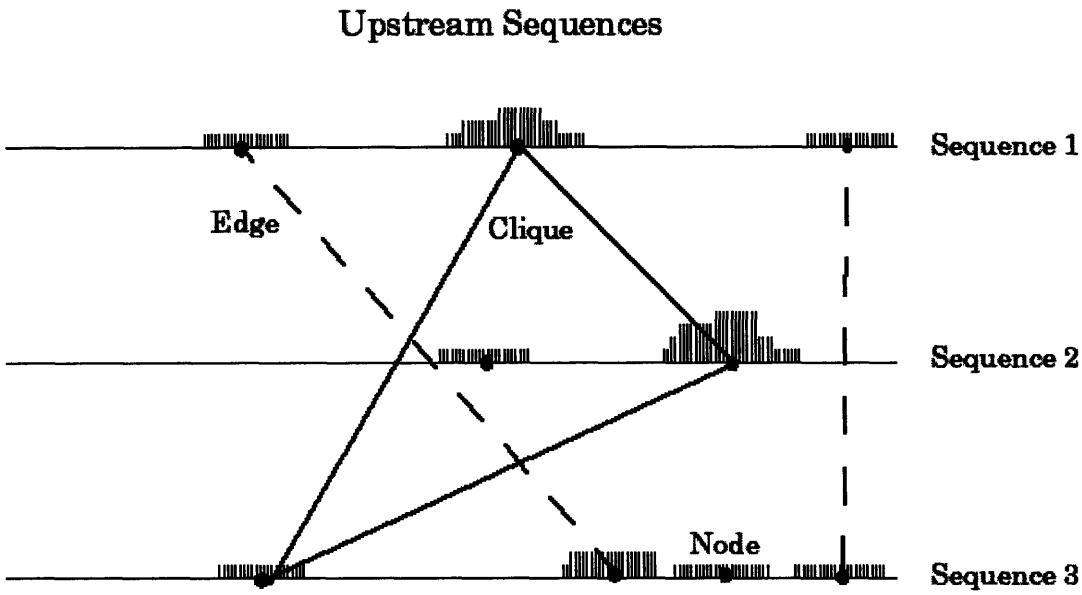


Figure 3-7: Clustering of similar sites in convolution phase: Shown is the feature map for three input sequences. A graph is constructed by representing each island/site with a 'node'. Edges are drawn between sites having high sequence similarity. Cliques in the graph represent clusters of highly similar sites, or motifs.

3.4.3 Ranking Motifs based on Significance

The final set of motifs found after clustering are aligned using the publicly available Consensus program developed by Stormo et al. [14]. The alignment is performed in two modes: (a) by including reverse strand sequences and (b) only with single strands sequences. By including the reverse strand sequences bias is given to motifs that are palindromic. Since this is a frequently occurring feature in binding sites (but not necessary) it is made optional. For a discussion about creation of alignments of binding sites see Appendix section A.5.

Consensus estimates a significance score of the alignment based on a large-deviation

method. The significance score or *p-value* is a measure of the likelihood of finding another alignment of the same "width" and "size" as the original alignment, in the set of input sequences with same or higher information content. Here, "width" of the alignment is the number of base pairs spanned by the alignment in width, and "size" is the number of sequences that constitute the alignment. When multiplied with the total number of possible alignments of that size and width that can be generated from the input set of sequences, one can obtain an *overall p-value* which can be used to compare the significance of two alignments having different widths and sizes (see equation 3.14). For a detailed discussion see [14].

$$\text{Overall p-value} = \frac{(T - m(l-1))!}{m!(T - m(l-1) - m)!} \times \text{p-value} \tag{3.14}$$

where T is the total number of base pairs in all input sequences, l is the width of the alignment in base pairs, m is the size of alignment and p-value is the significance score based on large deviation methods. Thus $T - m(l-1)$ is the number of starting positions for a random member of the alignment and the combinatorial term is the number of possible alignments of size m. In the rest of this dissertation, the overall p-value is referred to as the "P value". The motifs found are ranked on the basis of their P values and reported as the final output of the algorithm.

Chapter 4

Experimental Results

The performance of TABS was tested on known regulons from the well-understood *Escherichia coli* biological system. A set of 30 *E.coli* regulons, with known binding sites reported in literature, were used for this validation test. The idea of the tests was to assess the ability of the algorithm to find true binding sites. For reference, we compared the performance against two other popular algorithms, AlignACE and Consensus. In this chapter, we first provide an in-depth analysis of results obtained at various stages of the algorithm across the 30 test cases along with insights in cases where the algorithm shows high sensitivity to known sites, and others in which it does not. We demonstrate the ability of the algorithm to find motifs in cases, when unlike known regulons, *all* the input sequences do not share the same motif but instead the motif is present in a subset of the input sequences. We also demonstrate the ability of the algorithm to find "distinct motifs" in the same set of input sequences. Finally, we make novel predictions on the basis of the results from the 30 regulons.

4.1 Dataset of known *E.coli* Regulons

In order to test the performance of our algorithm it was desirable to select a biological system with reasonably well-understood regulatory mechanisms. *E.coli* being the most well-studied simple prokaryotic system, was the obvious choice. Several DNA-footprinting assays have been performed over the previous few decades on this system to identify DNA-protein binding sites. The results from these individual experiments are catalogued and continually updated in a publicly available database called RegulonDB [34]. Version 3.2 (2001) of this database lists regulatory information for 86 different transcription factors from *E.coli* including the information about the genes that they regulate, the transcription units involved in the regulation, the exact binding sites where the protein binds on the DNA. Besides the list of genes for which there *is* evidence of DNA-protein binding in literature, the dataset also includes a set of genes that conform to indirect regulation by the transcription factor, for which no binding site has been reported previously in the literature. Regulatory interactions in such cases are inferred from experimentally observed transcriptional changes [27]. For a detailed discussion about the experimental evidence on which the sites reported in RegulonDB are based, please see Appendix section A.4.

For the purpose of this study we chose 30 regulons from RegulonDB, where each regulon comprises a set of genes regulated by the same transcription factor. These regulons were chosen on the basis of two criteria : (a) there are at least three operons forming the regulon, and (b) there is at least one reported binding site in each regulon. A minimum size of three genes was chosen for each regulon in order to have a sufficiently large sample size for motif-discovery. Table 4.1 lists all the regulons used in this study. Also listed along-side is the number of operons in each regulon, the number of known binding sites in each operon, the number of genes with no reported binding site, the width of the reported binding site and the width of their consensus.

Regulon	Size	No. of Sites	No. of Genes with Indirect Regulation	Size of Reported Binding Site (bp)[a]	Width of Alignment(bp)
Ada	3	2	1	48	4
AraC	5	12	0	37	26
ArgR	7	12	1	36	26
CRP	73	103	15	39	24
CysB	5	4	1	62	16
CytR	7	13	1	60	46
DeoR	3	7	1	36	26
FadR	4	6	0	37	14
FIS	25	32	0	36	27
FNR	21	19	6	42	14
FruR	7	4	4	34	16
Fur	10	9	6	39	25
GlpR	4	17	0	40	18
IHF	22	20	9	33	16
LexA	9	9	1	40	20
Lrp	14	23	4	32	16
LysR	3	1	2	33	12
MalT	4	9	0	30	19
MetJ	3	5	1	28	18
MetR	3	3	1	44	16
NagC	4	6	0	46	29
NarL	12	15	4	39	16
NR-I	3	10	0	35	23
OmpR	6	14	2	30	8
OxyR	4	4	0	65	15
PhoB	5	11	1	37	25
PurR	17	15	3	36	16
SoxS	5	6	2	38	19
TrpR	5	5	0	47	28
TyrR	8	15	0	42	20

[a]Reported size includes 10bp flanking regions on either side

Table 4.1: Dataset of 30 known *E.coli* regulons from RegulonDB [34]

4.1.1 Building Sequence Logos

Binding sites recognized by the same protein as reported in RegulonDB, were aligned using the Consensus program. In certain cases, redundant sites were reported. These were manually removed. Complementary strand sequences were included in making the alignment except Ada and MetR regulons (see Appendix section A.5 for more details). The logos obtained are attached in Appendix section A-3.

4.1.2 Extraction of Upstream Regions

The complete genome sequence for the *E.coli* K-12 MG1655 strain was obtained from Genbank public database to perform the analysis. The chromosomal positions of the start codons for 4,405 *E.coli* genes, catalogued in RegulonDB, were used to extract upstream regions. As shown in Figure 4-1, a 400bp region upstream, plus 50 bp downstream from the reported start-codon, making a total of 450bp, were extracted for every operon. Of the total 419 sites bound by any one of the 30 transcription factors, 380 were found to be located within this 450bp region while the rest 39 ($\sim$ 10%) of them were found to be located farther upstream or even downstream of the transcription unit, typically $>$ 1000 base pairs away. Such sites were excluded from this study (except for the purposes of constructing sequence logos). The 50bp region downstream of the start codon is included to take care of any error due to reporting of start-codons[1].

4.2 Performance validation on 30 *E.coli* Regulons

Three metrics were used to evaluate the significance of results: (a) *sensitivity*, defined as the fraction of known sites correctly predicted, (b) *specificity*, defined as the fraction

[1]This is similar to the approach taken by Benítez-Bellón [2].

of *predicted* sites that hit known sites, and (c) a visual comparison of the sequence logos for the known motif and the predicted motif. The criteria for evaluating whether a known site is correctly predicted or not, is based on "touch". A known site is said to be "touched" by a predicted site if there is any degree of overlap between the two, when mapped on the sequences. Although, such a metric can overestimate the accuracy of prediction, when used together with the visual sequence logo comparison it works well. Another criteria used, calculates the overlap in terms of the actual number of base pairs "covered" (metric analogous to the metric used in section 3.3.1). This avoids the problem of overestimating, but since not all base pairs in a reported site form a part of the consensus, this metric can underestimate the accuracy of prediction. While the former metric is called "sensitivity by touch", the latter is termed "sensitivity by bps". In performance evaluation, we use both metrics in conjunction, although we use the former metric more often. Unless otherwise specified "sensitivity" refers to "sensitivity by touch" and likewise for specificity.

As seen in column 4 in table 4.1, several regulons have genes with *no* reported binding sites. This means that we have no way of deciding whether the predictions made on these genes are correct or not. This can lead to underestimation of specificity. To avoid this, we *correct* the specificity by basing it on the number of predictions for those genes that have at least one reported site. We call this the "corrected specificity". An explicit reference will be made when this metric of specificity is used; otherwise specificity simply refers to the usual definition which is based on *all* the predictions.

4.2.1 Summary of Results

The cumulative sensitivity of the algorithm across all the 30 regulons was found to be 40% and the cumulative specificity was 44.5%. The "corrected specificity" was

Category	No. of cases	Sensitivity %	Specificity(Corrected) %
Good Cases	14	68	83
Weak Cases	16	16	27

Table 4.2: Overall performance on the 30 regulons

53%. Figure 4-2 shows a histogram of the distribution of individual sensitivity and "corrected specificity". The distributions have a bi-modal nature.

In 14 regulons, the top predicted motif matched the consensus of known sites with a sensitivity and specificity (corrected) of 68% and 83% respectively. The consensus logos showed a high degree of agreement with the corresponding logos of binding sites. In the remaining 16 cases, either the algorithm found more interesting motifs (in terms of significance) than the binding motifs, or nothing significant was found. In the following sections we present the results at each stage of the algorithm.

4.2.2 Generation of Basic Pattern Set

All $l = 6, w = 20, k = k(n)$ (k is listed for different values of n in appendix section A.1.1) patterns were found using Teiresias. Table A.6 in appendix lists the number of patterns ($\sim 10^3 - 10^5$) found in each case along with the exact k values used. The number of top patterns selected on the basis of z-score, the "coverage" of these patterns on the input sequences, and the z-score of the least significant pattern selected, are listed in the same table.

As an example, consider the TyrR regulon containing 8 genes. The number of $l = 6, w = 20, k = 6$ patterns found by Teiresias when it was run with the 450bp upstream regions of these genes was $\sim 4,500$. The top 7 patterns are shown in table 4.3, aligned (for the sake of presentation), along with their z-scores. As can be judged from the alignment, the patterns appear to belong to a common motif.

Pattern 1	TGTMMWW...Y.TWKACA	z-score = 358.0
Pattern 2	C.WW.TGTMA.......TWKWC	z-score = 314.0
Pattern 3	TGTAMW....Y.WWTACA	z-score = 275.7
Pattern 4	TGKMMWW...Y.TWKACA	z-score = 253.9
Pattern 5	TGTMMW..WWT.WWKACA	z-score = 248.5
Pattern 6	AM....W.TG.AA..T....YW..C	z-score = 215.7
Pattern 7	TGKAMW....Y.WWTACA	z-score = 209.2

Table 4.3: Alignment of top 7 statistically significant patterns found in the TyrR regulon along with their z-scores

4.2.3 Feature Maps

Feature maps were plotted for each regulon. The islands identified were compared with the location of known binding sites and an "intermediate" sensitivity was computed. Figure 4-3 shows the feature map of the TyrR regulon obtained by mapping the 7 most significant patterns. Each line in the map represents the 450bp upstream region for each of the 8 operons in this regulon. The labels next to the lines indicate the name of the first gene in the corresponding operon. The leftmost co-ordinate of the line corresponds to -400bp (400bp upstream of the start codon), while the rightmost co-ordinate corresponds to 50 bp into the operon from the start codon and tics are placed at 100bp interval. At each position with a non-zero signal, one or more instances of the 6 patterns occur. The greater the number of instances that occur, higher the signal at that position. The bars shown below the line represent the locations of known TyrR binding sites. Some reported binding sites partially overlap with each other. There are a total of 10 islands, of which 8 overlap with known sites. The sensitivity is 73% and specificity is 90% "by touch", and 44% and 90% "by bps".

Category	No. of cases	Sensitivity (by touch)%	Sensitivity (by bps) %
Strong cases	13	83	62
Moderate cases	11	67	43
Weak cases	6	34	8

Table 4.4: Categorization of regulons on the basis of sensitivity at the end of filter 1 stage

The sensitivity and specificity for all the 30 regulons were computed up to this stage (Filter 1) of the algorithm (Table A.6).

On the basis of these results we could classify our results into three categories - (a) those with significant overlap between islands and known sites, (b) those with partial overlap and (c) finally those with minimal overlap (Table 4.4) . Figures 4-4, 4-5 and 4-6 shows the feature maps for these three categories respectively.

4.2.4 Results from Convolution

The final step of the algorithm, convolution, clusters similar islands into motifs. Table A.6 in the appendix shows the complete results for all the 30 cases, including the number of motifs found by clustering, the P value of the *best* predicted motif, the width of the motif found, and the sensitivity and specificity of the motif.

The cases categorized as ones with "minimal overlaps" based on results from statistical filtering, are not expected to give any fruitful results in the convolution phase of the algorithm. Among the cases with partial overlaps, 3 were found to give good results, and among the "strong cases" 11 were found to give good results. Together, these 13 cases formed the "good cases". The sensitivity and specificity for these cases along with the P value of the best motif are reported in table 4.5. The average values for sensitivity are "corrected specificity" across these 14 cases is 68%,

Regulon	p-value	Sensitivity (%)	Specificity (%)	Specificity %(corrected)
ArgR	-209.55	83	71	99
CRP	-391.68	49	45	54
CysB	-104.79	50	40	50
DeoR	-67.11	50	50	67
FruR[a]	-45.11	75	60	100
Fur	-338.37	67	25	60
GlpR	-87.94	40	100	100
LexA	-120.77	78	88	101
MetJ	-62.15	100	67	101
NR-I	-36.54	56	100	100
PhoB	-177	71	83	83
PurR	-218.01	85	52	57
TrpR[a]	-55.67	80	100	100
TyrR	-173.22	73	91	91
	Mean:	68	69	83

[a]2nd best motif

Table 4.5: Sensitivity and specificity for the 14 "good cases".

and 83%, respectively. The sequence logos of predicted motifs showed an excellent match with the sequence logos of known motif (Figure 4-7). The remaining cases showed much lower sensitivity and "corrected specificity" of 16% and 27% respectively and the predicted motif (if any) did not match the consensus of known sites. These cases are discussed and analyzed in detail in section 4.2.5.

4.2.5 Analysis of Weak Cases

In this section, we present the results and analysis for the cases in which the algorithm failed to identify the real binding sites. The discussion is treated by dividing the cases into three categories :

(a) More significant patterns found: In these cases the known motifs were too degenerate. As a result, Teiresias either found more significant patterns than those corresponding to the known motifs filtered out, or did not find any pattern corresponding to known motifs.

Consider the Ada regulon. This regulon consists of three promoter regions corresponding to the genes alkA, ada and aidB. RegulonDB reports two binding sites for this transcription factor, one each in the promoter region of alkA and aidB. The alignment of these two sites is shown in Figure 4-8. When Teiresias was run with $l = 6, w = 20, k = 3$ on this regulon, it returned 27,601 patterns. One of the patterns, `A.MRRAAT.W.W.MGCAA`, having a z-score of 149.4, contained the binding-site consensus pattern `A....AAT......GCAA`. However, the top 5 patterns selected, had a much higher z-score of 1256405.6 and above. (Figure 4-8). Thus the discovered patterns (a) were much more significant than the binding site, and (b) had nothing to do with the binding sites. Such a behaviour was found in 8 other cases listed in table 4.6. Column 2 lists the best pattern discovered that contained the known binding site, column 3 lists the corresponding z-score and column 4 lists the z-scores of the selected patterns.

The findings reflect two possibilities. One possibility is that the regulon size is too small to identify the correct signal. In other words, one would need sequences with more instances of occurrence of the known motif for effective motif-finding. In fact, from a z-score study on random sequences of same size as the individual regulons, we estimated the significance values of the reported z-scores. It turns out that for OxyR, OmpR and CytR these significance values are < 0.05 (data not reported), implying

Regulon	Size	Pattern containing consensus of known sites	Support of the pattern	z-score	z-score of patterns selected (>=)
Ada	3	A.MRRAAT.W.W.MGCAA	3	149.4	1256405.6
CytR	7	Not found	-	6	320.7
FadR	4	A..WG.TC.G..Y.....T	4	57	3657
MalT	4	C.S.RGGWKGAG.W.....MT	4	315.0	3700.8
MetR	3	Not Found	3	-	7253791.7
NagC	4	Not Found	4	-	20991.8
OxyR	4	A.Y.R..R.YATR....ATC.Y..Y.AT	4	1128.5	7057.2
OmpR	6	Not found	5	-	824.1
SoxS	4	Not found	4	-	2606.6

Table 4.6: Z-scores of patterns corresponding to known sites are much lower than those of the z-scores of top patterns selected. In some cases, no such patterns were found due to degeneracy of binding sites

that the motifs found may have some biological significance. For the remaining cases, it is likely that the motifs are an artifact of the data.

(b) No clear signal found: These are cases in which top patterns mostly consist of *poly A's* and *poly T's*. These patterns were checked for any possible overlapping occurrences in the genome which could lead falsely lead to over-estimation of their z-scores. We did not find any overlaps for these patterns.

Consider the example of NarL. The best pattern found by Teiresias in this regulon of size 12 is A....YTMW..MAA..A..A....A with a z-score of 104.6. The sequence

Regulon	Best pattern found	z-score
NarL	A....YTMW..MAA..A..A....A	104.6
Lrp	T..W...TTT.K.T....MT....	86.8
IHF	A.....W..TT.A..TT.....TT	82.7
FNR	T.....T....ATTWA.....KWT	71.5

Table 4.7: Cases with weak signal - mostly poly A's and poly T's found

logo for this pattern is shown in Figure 4-9. Clearly this contains "poly A" which is ubiquitous all over the genome. The alignment corresponding to known sites is shown adjacently in the Figure. This is also very weak. A simple search shows that the consensus T.....TA.....A occurs $\sim 7,400$ times in all the upstream regions of *E.coli.* It could be surmised that in these cases no clear "sequence signal" exists and that perhaps there are features beyond primary sequence, such as the secondary structure of the DNA, that determine how the protein recognizes such a site. Table 4.7 provides a list of the 4 cases fitting in this category along with the most significant pattern found.

(c) Special Cases: In two other cases, namely AraC and FIS, we observed that even though the consensus of known sites was reasonably good, our algorithm could not find them. This was due to complex degeneracy in binding sites.

Consider the example of FIS. This regulon has a size of 25 with 32 reported binding sites. The consensus for these sites is shown in Figure 4-10. When Teiresias is run on this set of sites with $l = 6, w = 20$ and $k = 9$, one pattern is found containing the consensus TG...A.....T...CA (z-score=8.8) in 9 sequences, and 39 patterns are found containing TG.....CA (3.7 <z-scores< 24.8) each with a support of

10. Although individually none of these patterns targets all the binding sites, together they touch all 32 of them. Since none of these patterns have high enough z-scores to make it to the top they all get filtered out. The patterns that get selected have z-score > 939.6 and have no similarity with binding sites. Notably, these patterns have a support of only 9. Hence they occur in only a small proportion of the input sequences.

This is an example of a case where signal enhancement takes place by mutual reinforcement from a collection of weak patterns.

4.3 Comparison with other Algorithms

The performance of TABS was compared against two other popular algorithms - AlignACE and Consensus. Performance was evaluated on the same dataset described earlier in this chapter in section 4.1. Academic licenses for AlignACE and Consensus were obtained from their respective authors and the software programs were downloaded and run locally.

AlignACE was run with following settings:

number of columns to align (10)	-numcols	=	default (10)
number of sites expected in model (10)	-expect	=	number of sequence in the input set
background fractional GC content of input sequences	-gcback	=	0.24

The remaining parameters were kept to default settings. AlignACE searches on both forward and double strand and returns a list of motifs sorted by their MAP (maximum *apriori* likelihood) score. AlignACE returned more than 50 motifs for each of the 30 regulons. Only the first motif (having the highest MAP score) was used for analysis. The sensitivity and specificity (by touch) of the results were computed. The detailed results for each regulon are provided in appendix table A.8.

Consensus was run with following settings:

```
Number of standard deviations for identifying information peaks   -s    = 1
Ascii alphabet information on the command line                   -A    a:t 0.26 c:g 0.24
Number of final matrices to print                                -pf   1
Ignore the complementary strand (default)                        -c    ,0 or
Include both strands as separate sequences                       -c    1
```

Consensus was run in two modes, once with the -c0 option (complementary strand ignored), and other with -c1 (complementary strand included) option. We denote the first mode by "Consensus-single" and the second mode by "Consensus-double". The sensitivity and specificity were computed using the motif returned from the final cycle. Results are listed in table A.7 (for Consensus-single) and table A.6 (for Consensus-double).

A comparison in performance has been drawn out in between the three algorithms in table 4.8. algorithm.

AlignACE has a very low specificity of 10% which means it returns too many false positives. This can also be judged from column 4 which lists the mean number of sites per gene as 8, unreasonably big. The sensitivity of 49% becomes meaningless in light of the low specificity. Consensus-double finds extremely long motifs in four regulons: AraC (137bps), GlpR (91bps), IHF (89bps), LysR (172bps) and NagC (174bps). These falsely create high sensitivity and specificity figures (see table A.6) and were hence assigned zero sensitivity and specificity in all the calculations shown. While the motif corresponding to CRP regulon is also long (75bps), it captures the correct consensus (data not shown) and hence is retained as is. TABS outperforms all the algorithms. Column 4 highlights the fact that Consensus assumes one site per gene in its algorithm, while TABS does not have this restriction.

When performance is compared individually, we find that Consensus also shows a bimodal distribution of sensitivity and specificity. In fact, in 15 of the 16 cases in which TABS does not perform well, Consensus does not perform well either (the 16^{th} case is CytR). On the other hand, there are 4 cases (PhoB, GlpR,DeoR and CysB)

Algorithm	Mean Sensitivity	Mean Specificity (corrected)	No. of Sites per Gene
TABS	40	53	1.2
Consensus-single	31	38	1
Consensus-double	33	32	1
AlignACE	49	10	7.9

Table 4.8: Comparison of performance of TABS with Consensus and AlignACE on the 30 *E.coli* Regulons

in which Consensus fails but TABS manages to find the correct motif.

Similar comparison is drawn between the algorithms, this time using only the 14 "good cases" plus CytR case (on which Consensus performs well), This is done to obviate the effect of any noise due to the remaining cases on which none of the above algorithms perform well and are just "weak cases". TABS still emerges with the best performance (table 4.9).

Algorithm	Sensitivity	Specificity (Corrected)
TABS	64	78
Consensus-single	53	69
Consensus-double	60	76
AlignACE	56	13

Table 4.9: Comparison of performance in those cases on which at least on algorithm performs well (14 "good cases" and CytR)

4.4 Performance on Synthetic Microarray Data

Since the ultimate application of this algorithm is to find conserved sequences in set of genes that result from clustering of microarray data, several tests were conducted to demonstrate the performance on such kind of data. Clusters of co-expressed genes obtained from microarray experiments are noisy which it makes it harder to find a conserved motifs in the corresponding upstream sequences. For instance, from a set of 15 coexpressed genes only 10 or fewer of them might share the same regulatory element. Our algorithm has several features that make it suitable for use on such kind of data. Firstly, the value of k chosen is much smaller than the number of sequences in the input set. This makes it possible to capture patterns conserved in a only a small fraction of input sequences. Also if there are multiple "regulons" that exist within the cluster, our graph-theory based clustering algorithm would enable us to capture each one of these as distinct motifs.

In order to test the performance, "microarray-like" data was created. Gene clusters were "synthesized" by grouping genes in a particular regulon with *random* upstream regions from the genome. The regulons chosen were the ones in which a clear signal had been identified in the experiments decided earlier in this chapter. Two regulons were chosen for this study: LexA (9 genes) and PurR (17 genes). Each regulon was mixed with random sequences in varying proportions and inputted into TABS. The effect on sensitivity and specificity was examined. The experiments and results are summarized in Table 4.10. Also shown alongside, are results from Consensus on the same datasets.

The sensitivity and specificity are stable even after introduction of spurious sequences (there is in fact an increase in LexA sensitivity possibly due to increase in size that makes it easier to identify discriminatory patterns). It must be mentioned here, that the best cut-off for number of top edges to be selected in the clustering

stage were obtained after a trial and error procedure. The performance is robust for sensitivities of up to 30%. The performance of Consensus was found to be similar to TABS in these cases.

Regulon	Original Size	Original Sensitivity	Orig Specificity	Spurious Genes	Total No. of Genes	Fraction of spurious sequences	TABS		Consensus	
							Sens	Spec	Sens	Spec
PurR	17	92	55	3	20	0.15	85	59	91	60
	17			4	21	0.19	82	51	91	57
	17			5	22	0.23	91	57	90	54
	17			6	23	0.26	91	57	92	52
	17			7	24	0.29	91	54	91	50
LexA	9	78	88	3	12	0.25	89	55	89	58
	9			4	13	0.31	89	80	89	54

Table 4.10: Table showing performance on synthetic microarray data

Besides the situation emphasized above, where only the genes in a subset of the gene cluster share a common motif, it is also possible that two *distinct* motifs are shared among two subsets of the original gene cluster. This kind of a situation occurs frequently in microarray expression data. Consider, for instance, a situation in which transcription factor A induces a set of genes a,b,c and d. Gene a encodes for another transcription factor (say, B) that in-turn induces a set of genes e,f,g and h. Thus genes a-d share a common regulatory element in their upstream regions, while genes g-h share another motif. It is entirely possible that all these genes (a-h) have similar profiles in a time-series experiment or under different sets of experimental conditions, and hence get clustered together. Unlike most conventional algorithms, TABS is designed to account for this possibility as shown in the experiment done below.

To illustrate the above scenario, an experiment was performed in which genes from two different regulons were "mixed" to form a heterogeneous cluster. TABS was run on this heterogeneous cluster to see if it could find the two motifs corresponding to the two regulons, separately. The results were compared with those from Consensus.

LexA (9 genes) and TyrR (8 genes) regulons were mixed and the total of 17 genes were inputted into TABS. TABS reported a total of 9 motifs. The first motif corresponded to the LexA binding site motif, while the 9th corresponded to TyrR (see table 4.11 and Figure 4-11). Top 10 motifs from the final cycle of Consensus were tested for matches to the LexA or TyrR motifs. None of them was found to be specific to either LexA or TyrR.

Algorithm	Sensitivity wrt LexA	Specificity wrt LexA	Sensitivity wrt TyrR	Specificity wrt TyrR
TABS - Motif 1	89	80	0	0
TABS - Motif 9	22	22	47	78
Consensus	89	47	22	20

Table 4.11: Ability to pick two distinct motifs from a set of genes: LexA and TyrR regulons were mixed to form a cluster of size 17. Among the 9 motifs found, motif 1 was found to resemble LexA site, and motif 9 was specific to TyrR sites. Consensus could not detect a motif specific to either of the two sites.

4.5 Novel Predictions

The 14 cases in which the algorithm managed to find correct motifs were used to make novel predictions. The predictions were made at two levels. The first set of predictions was made on those genes in the regulon that had no reported binding site. Second set of predictions were made at the genomic scale, i.e. in the upstream regions of all the genes in the genome.

Before making these predictions, a post-processing step was performed on the predicted motifs to "clean-up" possible false positives or false negatives that could

be artifacts of various thresholds used in the algorithm. Essentially, this involved constructing a PSWM of the predicted motif, and sliding the PSWM along a set of sequences to find possible high-scoring matches. For this purpose the Matind and Matinspector package [25] was used. Given a motif or a set of aligned sites, Matind constructs a PSWM which is used by Matinspector to find matches in a given set of sequences on the basis of a specified "matrix-similarity" threshold.

Using these tools, the "clean-up" step was done in two steps. As a first step, the threshold of matrix similarity was determined by scoring each site that was used to construct the PSWM itself. If a site was found to have a score much lower than the remaining sites, it was excluded from the list. A suitable threshold deduced from the scores of the remaining sites was determined and used while scanning an arbitrary set of sequences to find possible matches using Matinspector. Columns 4 in table 4.12 lists the matrix-similarity thresholds obtained for the 14 regulons using this procedure.

For the first set of predictions, the PSWM was scanned against the upstream sequences of all the genes in the regulon. By doing this we found some new sites in certain cases that were missed earlier. Also some spurious sites were removed. This usually led to improvement in either sensitivity or specificity or both (table 4.12). The number of novel sites found in each regulon, and the threshold of matrix similarity are also listed in the table. (Note: 2 cases CRP and Fur were not considered in this analysis since their matrices were found to be fairly non-specific.) Figure 4-12 shows a map with the predictions in the PurR regulon as an example.

The first 8 regulons listed in the table having very high sensitivity and specificity, were used to make genome-scale predictions (using the same matrix-similarity thresholds). See table 4.13. The exact locations of these sites is documented in the appendix.

Regulon	Sensitivity	Specificity	Matrix Similarity Threshold	Number of Predictions
ArgR	0.83	0.83	0.85	1
FruR	1.00	0.57	0.88	3
LexA	0.89	0.89	0.85	1
NR-I	0.67	1.00	0.85	0
PhoB	0.71	0.83	0.85	0
PurR	0.92	0.93	0.85	6
TrpR	0.8	1.00	0.9	0
TyrR	0.8	0.92	0.81	1
CysB	0.5	0.5	0.85	3
DeoR	0.5	0.67	0.94	1
GlpR	0.4	1.00	0.9	0
MetJ	1.00	0.67	0.95	1
			Total Predictions	17

Table 4.12: List of first set of predictions in genes included in the regulon.

Regulon	Number of Predictions
ArgR	7
FruR	112
LexA	103
NR-I	9
PhoB	0
PurR	320
TrpR	2
TyrR	361
Total Predictions	914

Table 4.13: Predictions at the genomic scale.

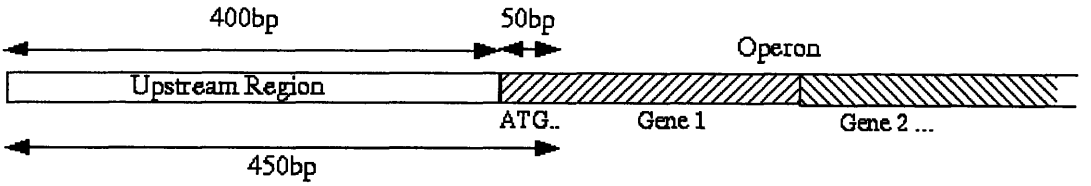


Figure 4-1: Extraction of 450bp upstream regions

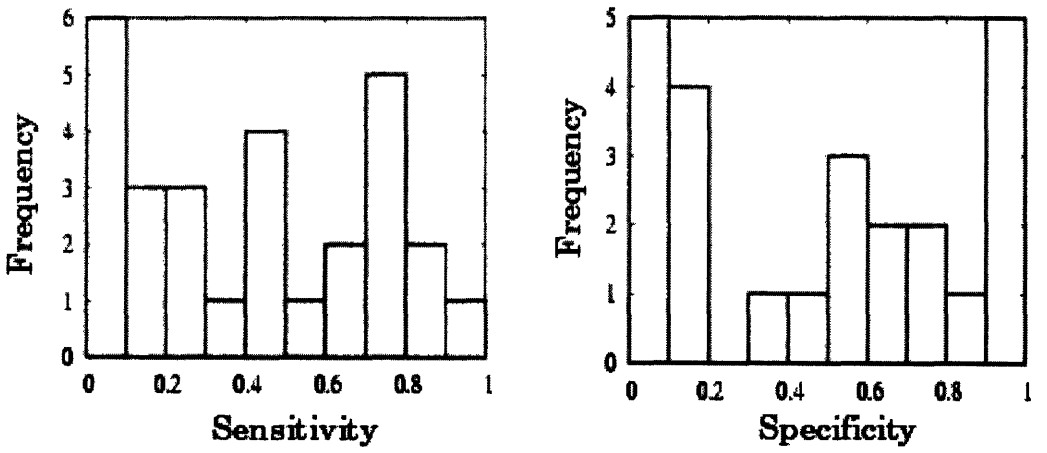


Figure 4-2: Histogram of sensitivity and specificity across 30 *E.coli* regulons: both plots reflect a bi-modal nature

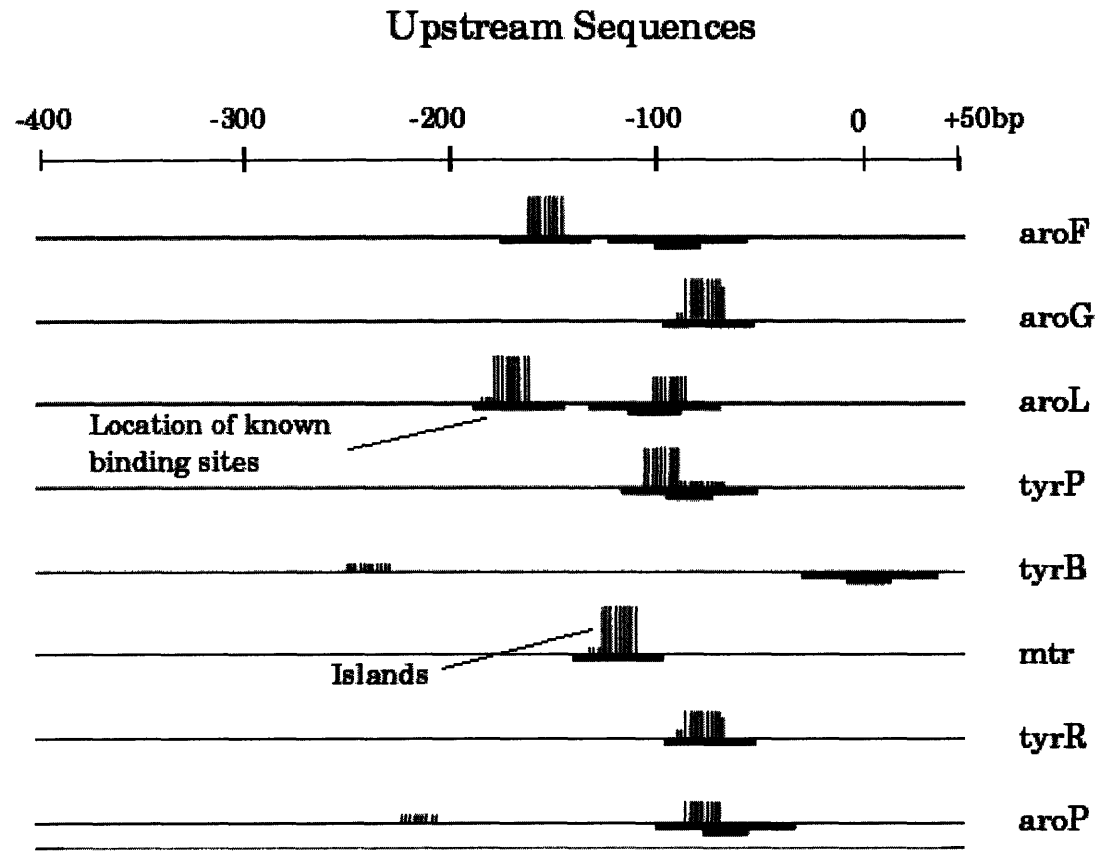


Figure 4-3: Feature map for TyrR regulon example showing overlap of significant patterns with known sites

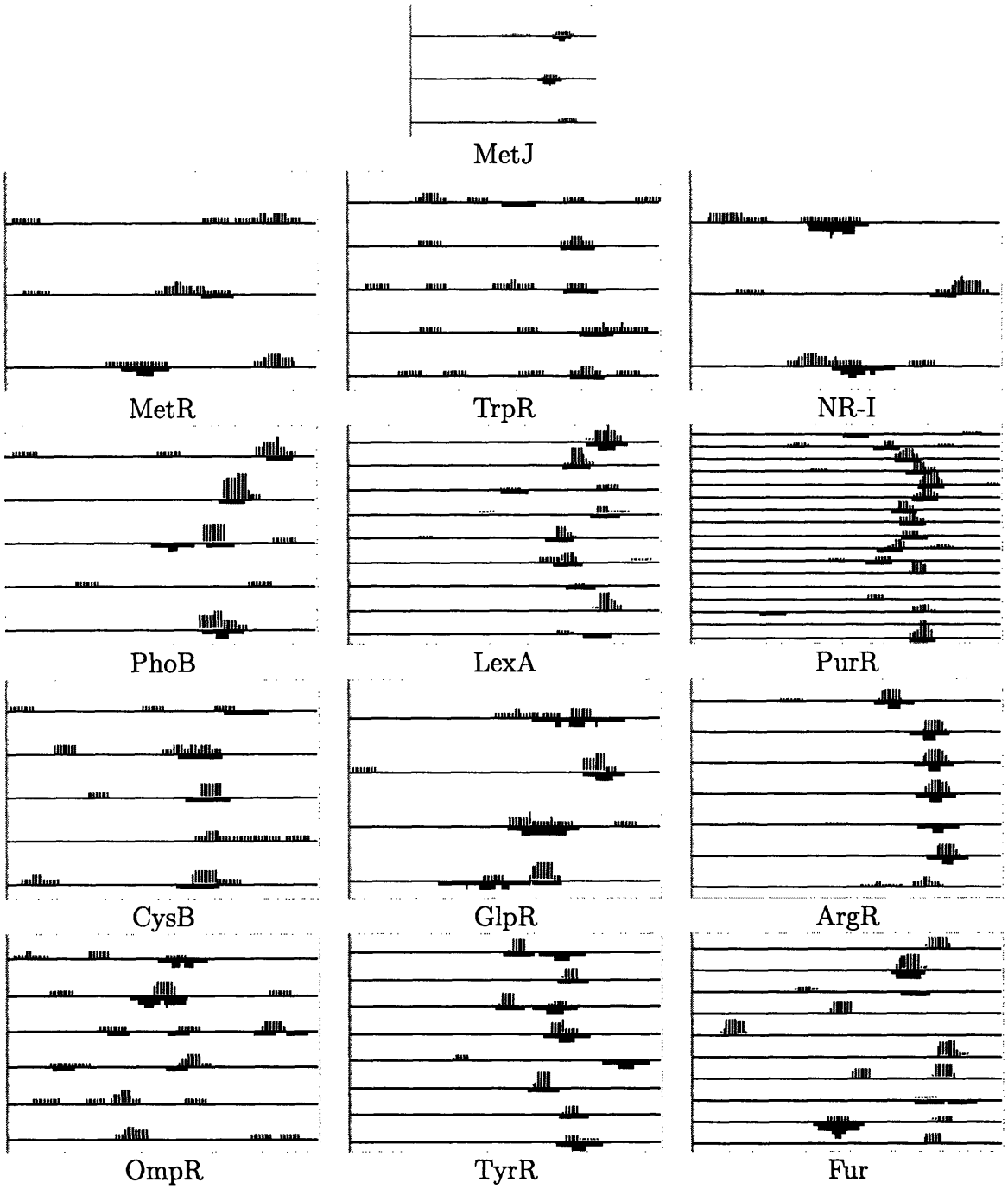


Figure 4-4: Feature maps for 13 cases showing high degree of overlap between patterns with high significance and binding sites

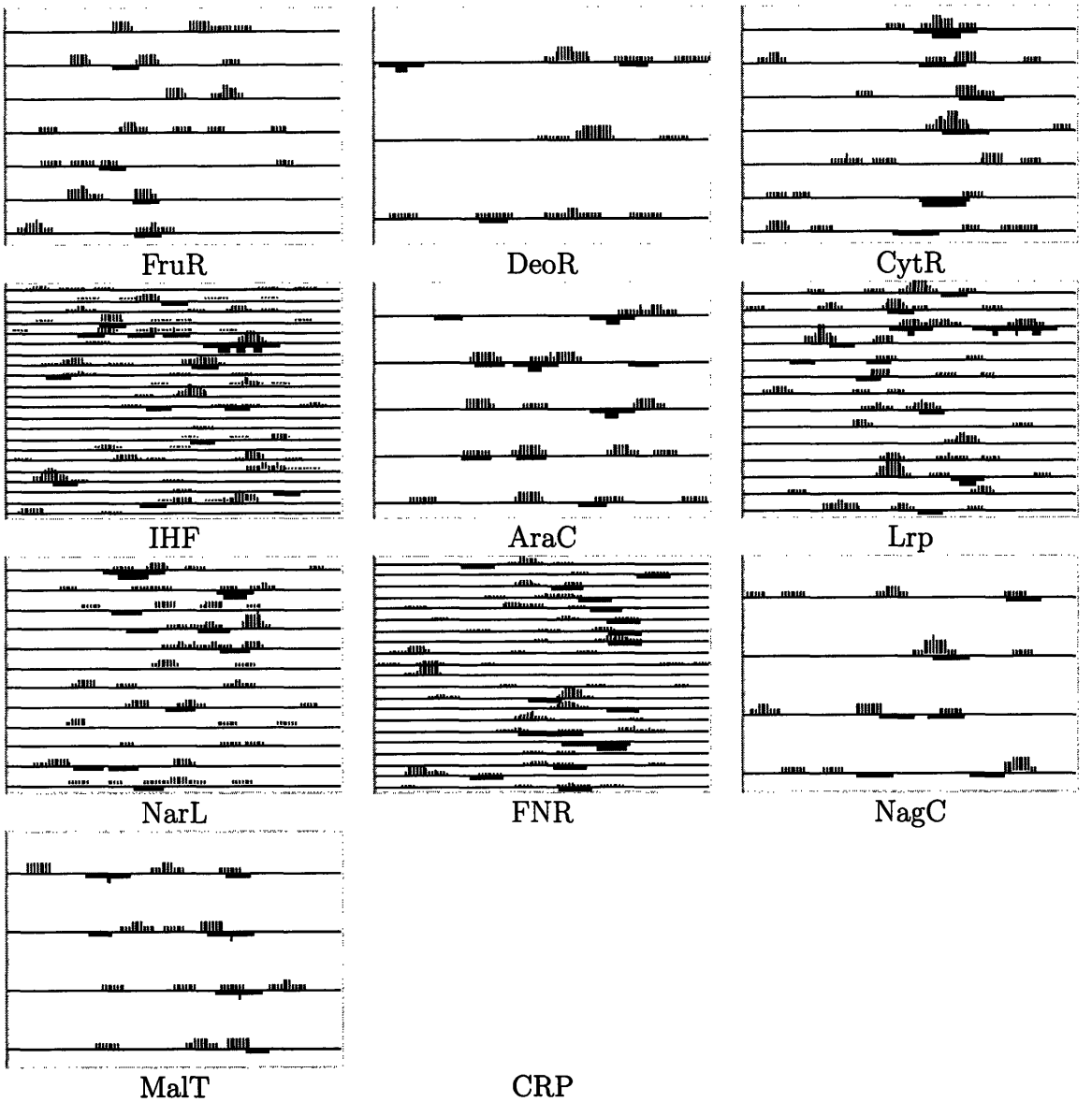


Figure 4-5: Feature maps for 11 cases with partial overlap between significant patterns and binding sites. The map for CRP has not been shown because of the huge size of the regulon making it difficult to depict graphically.

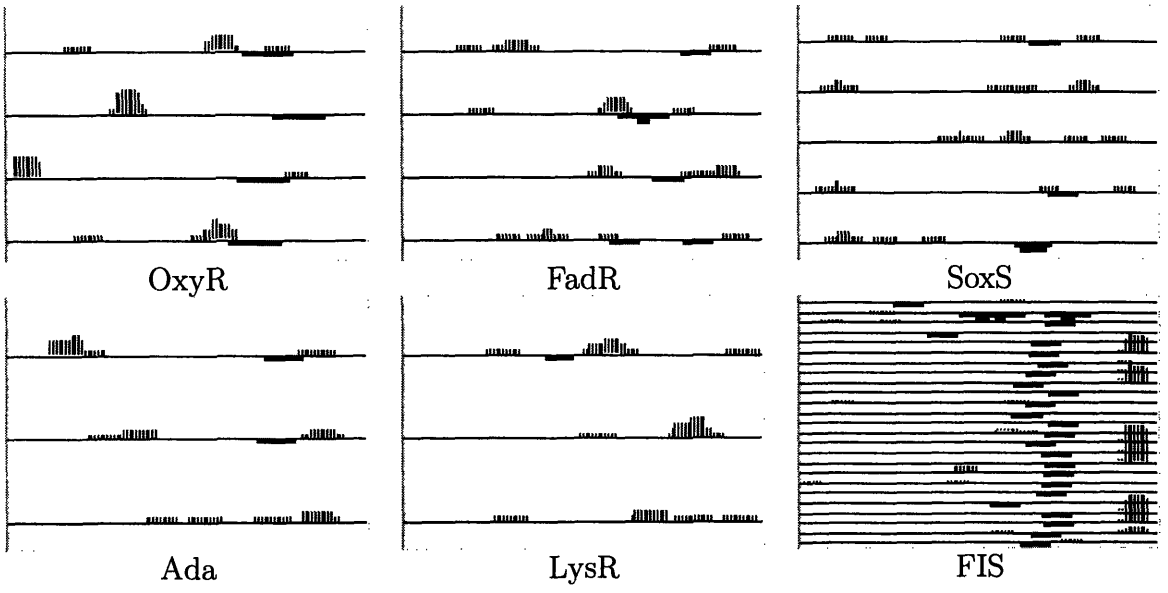


Figure 4-6: Feature maps for 6 cases with very little overlap between significant patterns and binding sites

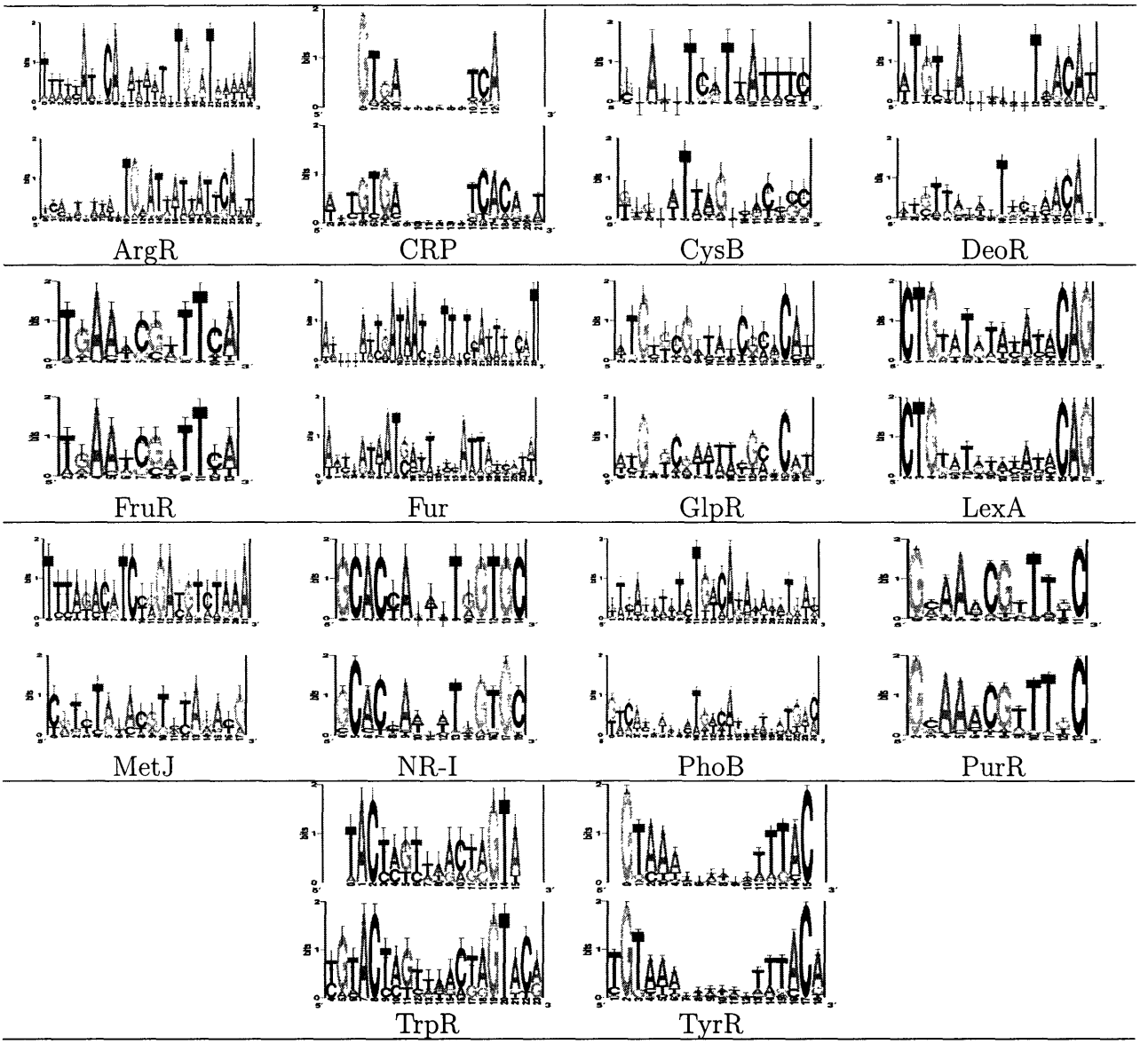
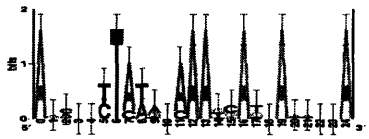


Figure 4-7: Comparison of predicted motifs with known motifs for 14 cases. For each regulon the motifs are aligned vertically with the predicted motif on top and binding site motif directly below it. In 2 cases MetJ and ArgR the predicted motifs look different from the corresponding logos of known sites, even though the sensitivity and specificity are high. This is because of the overlapping occurrences of binding sites *adjacent* to each other in all the upstream regions in these regulons (see feature maps for MetJ and ArgR in Figure 4-4). Our algorithm tends to find maximal patterns and hence reports one motif corresponding to alignment of the entire stretch of sites.

Pattern corresponding to known consensus	A....AAT.....GCAA	z-score = 149.4
Least significant of the 5 patterns selected	AWSS.RKYG.CC.W.SKAAYRS.R.CTGS.WTYSRTSR.SG	z-score = 1256405.6

Figure 4-8: Ada case to exemplify situations where more significant patterns than the consensus of known sites are found



Alignment of best pattern found(z-score=104.6)

Motif from alignment of known sites

Figure 4-9: NarL case to exemplify situations where there is no clear signal found in the upstream regions

Figure 4-10: FIS binding-site motif: FIS case to exemplify special situations where the algorithm fails to identify the motif

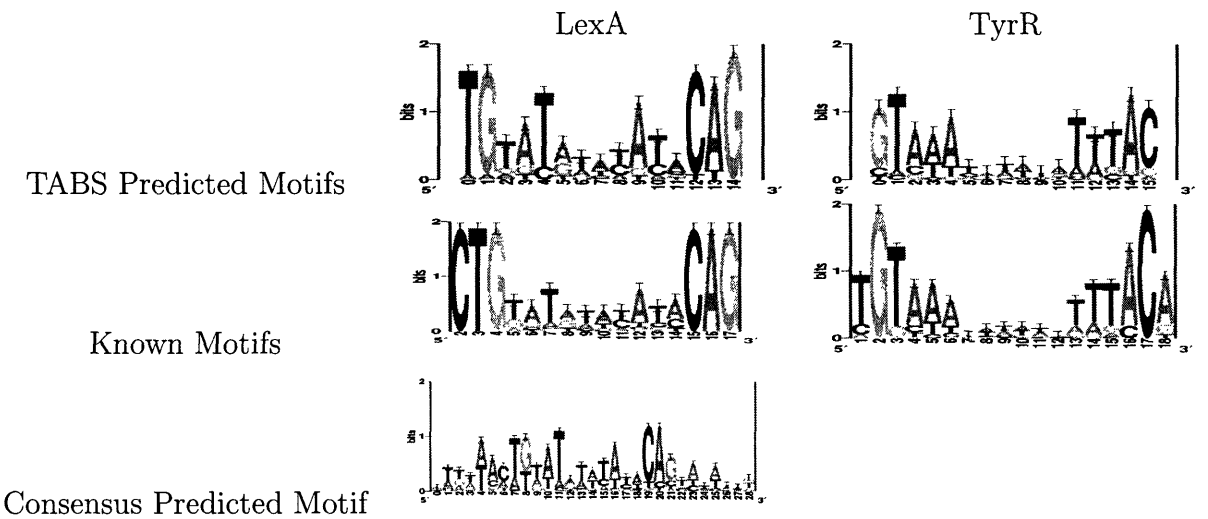


Figure 4-11: Logos for motifs found in the "mixed" regulon

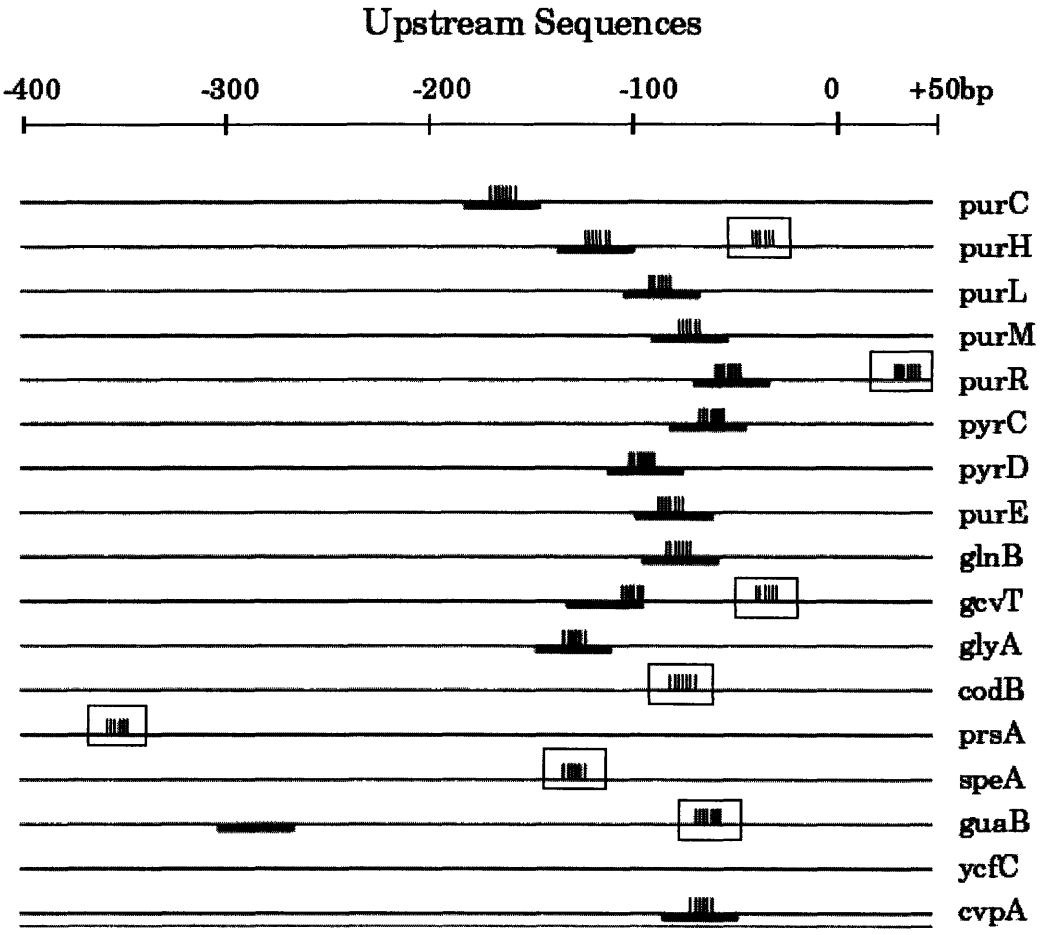


Figure 4-12: Novel predictions in the PurR regulon: Boxed sites represent novel predictions. Previously reported binding sites are shown by the solid bars.

Chapter 5

Discussion

5.1 Summary

Promoter sequences encode a wealth of knowledge about gene regulation. Unlocking this information is key to developing an understanding of how the cell operates in response to various external stimuli and environmental factors. Identifying promoter sequences at a genomic scale using sequence information is still largely an unsolved problem. There is much to be understood about how DNA-protein interactions occur and how binding sites recruit transcription factors for turning genes on and off. Several attempts have been made in the past to discover binding sites by exploiting sequence characteristics of these motifs. Most of these approaches can be categorized as either *sequence-driven* or *pattern-driven*, each having its own strengths and limitations. Sequence-driven methods provide better mathematical descriptions of motifs using weight matrices, but cannot guarantee optimal solutions. Pattern-driven approaches only search for special types of motifs that are usually not sufficient to model the entire spectrum of binding sites found in nature. In this thesis, we propose a novel method, called TABS, that integrates the advantages of both these approaches. We

make use of Teiresias, a pattern discovery algorithm that finds all exhaustive and maximal patterns from a given set of sequences. Teiresias has the ability to find patterns of variable length using its unique algorithm that ensures maximality. It is a fast and efficient algorithm that generates results in a time-scale linear with respect to the output size. Since the search is exhaustive, theoretically speaking no possible solution can be missed (unlike sequence-driven approaches), and the type of patterns discovered belong to a much wider pattern class than those found by most pattern-discovery algorithms. We use this algorithm to enumerate a basis set of patterns that describe conserved regions in the input sequences. Since motifs can be vary highly and be degenerate, choosing the right parameters for pattern discovery is important, and the best choice can vary from case-to-case. The parameters $l = 6$, $w = 20$ were found to give most sensitive results when tested on known *E.coli* regulons. k, the minimum support of each pattern, is usually set much smaller than the number of input sequences. Use of rigorous statistical criteria enable selection of significant patterns that are most likely to constitute a motif. These patterns are used to delineate "interesting" regions in the input sequences. Thus, using pattern enumeration the problem is reduced from finding motifs in large chunks of nucleotide sequences, to one of finding motifs in a much smaller and shorter set of sequences. Sequence-driven-type approaches are then employed to find shared motifs in these regions by using simple pairwise similarity criterion to group sequences on the basis of their information content.

We tested the performance of TABS on 30 E.coli regulons by evaluating its ability to find experimentally proven binding sites. In 14 of the 30 cases a high sensitivity of $\sim 70\%$, and specificity greater than 80%, was obtained. The corresponding sequence logos showed high degree of visual similarity. In the remaining cases, either more interesting patterns, than the ones reported, were found, or low complexity regions such as poly A's and poly T's were found. TABS was found to perform better than

two other state-of-the-art algorithms - AlignACE and Consensus. Using synthetic experiments, the ability of the algorithm to find sites existing only in a small subset of sequences in the input set was demonstrated. This is especially relevant for analysis of coexpressed gene-clusters obtained from DNA microarray experiments.

5.2 Future Work

1. *Improved pattern discovery model*: Teiresias is limited in its search to find patterns with rigidly conserved positions Thus, it fails to detect motifs having excessive degeneracy such as those represented by the binding sites of FIS and AraC. This is a common problem with most pattern-driven motif-finding algorithms. By enumerating patterns with a low level of support, this problem was partly offset in the algorithm developed. However, in order to address this issue completely, there is a need to develop pattern-discovery approaches that allow degeneracy in matched instances. Along these lines, there is an ongoing effort in the Bioinformatics and Metabolic Engineering group at MIT to develop an improvised pattern discovery tool that constructs patterns using *alignments* scored on the basis of "scoring matrices", rather than *exact matches.*

2. *Application and testing on microarray data*: A limited set of synthetic experiments were performed in this study to evaluate the performance of the algorithm on microarray-like-data. It would be useful to extend this work by demonstrating the performance of the algorithm on real microarray datasets and compare the findings with other algorithms. Such experiments can also provide an extensive list of targets for biological validation in laboratories.

3. *Detection of Composite motifs*: Gene regulation is observed to occur via combinatorial arrangement of regulatory motifs in the promoter regions. This is

especially true for higher eukaryotes. From a statistical perspective it is often easier to locate a *group* of regulatory motifs together (since the probability of occurrence of that event is much lower), as compared to discovering *individual* motifs. This can be facilitated by constructing pattern classes that specifically model for composite motifs using flexible gaps and spacers. Such models can be very complicated and diverse. An excellent review of composite-motif-based methods for finding binding sites has been documented by Sinha [40]. While Teiresias does not handle flexible gaps currently, there is an ongoing effort to add this functionality [28].

4. *Genome-scale motif detection*: Another attractive approach for motif-finding is to search the entire sequences of upstream regions in the genome, altogether in a single stretch, looking for conserved elements in small subsets of these sequences. While sequence-driven approaches are computationally too expensive for such kind of a search, there have been various pattern-driven approaches used in the past for genome-scale discovery [21], [7]. The pattern class used in such cases tends to be very restrictive because of space complexities involved, but certainly not impossible. Indeed, use of Teiresias for full-genome pattern finding is complicated for similar reasons of computational time and space complexity. An additional challenge that needs to be addressed is choosing the right set of parameters, given that there is little known about the structure and size of "regulons" at a genomic scale.

5. *Cross-species comparison*: Another approach, increasingly gaining importance, is based on searching motifs in upstream regions of *orthologous* genes across several different, but related species. The underlying hypothesis is that, through the process of evolution, vast chunks of biologically insignificant portions of non-coding genomic regions would have mutated considerably, while the functionally

active regions would be conserved. Such a phylogenetic comparison has shown promise in the recent past [24].

5.3 Contributions

1. The main contribution of this thesis is the development of a novel hybrid-approach that combines the pros of *sequence-* and *pattern-driven* approaches making it superior to most existing algorithms conceptually. To this end, the algorithm leverages Teiresias, by using it in a manner that makes it suitable for discovering binding motifs. This includes determining the appropriate values of l and w, and making a proper choice for k, which when integrated with the "mapping" concept, facilitate assembly of *degenerate* motifs.

2. Statistically significant patterns, obtained from stage 2 of the algorithm, were shown to demonstrate high sensitivity towards binding sites.

3. A new alignment procedure for constructing motifs of binding sites involving inclusion of reverse strands was developed. The method showed the ability to detect signals that were otherwise not apparent using conventional alignment techniques.

4. A method for clustering sequences by finding cliques in a graph constructed using an *all-against-all* alignment approach, was developed. Unlike conventional techniques, this method can find multiple clusters without the need to specify the number of clusters in advance.

5. Detailed studies on 30 *E.coli* regulons provided insight into the varying types of binding motifs found in nature, which further provide leads for development of better motif-finding tools in the future.

6. The algorithm was shown to perform better than two other popular algorithms, Consensus and AlignACE.

7. A set of novel predicted sites was documented for 12 transcription factors in *E.coli* based on the results found. These sites show high degree of similarity with the motif recognized by the protein and this compelling evidence makes them excellent candidates to be tested in laboratories.

5.4 Conclusion

This thesis has addressed the problem of finding transcription factor binding sites with considerable success, and at the same time has provided some insights into the problem which have opened new avenues for further research. In particular, a novel method has been developed, that addresses some the shortcomings of existing promoter-finding algorithms. The addressed problem is of tremendous significance in our understanding of interaction between genetic information and cellular function.

Appendix A

Appendix

A.1 Algorithm Parameters and Thresholds

A.1.1 Support (k) vs. n

$i <= n <= j$		
i	j	k
3	3	3
4	5	4
6	6	5
7	9	6
10	12	7
13	20	8
21	30	9
31	40	10
41	50	11
51	60	15
61	100	20

Table A.1: n is the number of sequences

A.1.2 Statistical Filtering Threshold

We wish to estimate the number of patterns that can provide a mean coverage of 20% when mapped on the input sequences. Let the number of input sequences be n. We assume the average length of a pattern is 20bps, and the length of each input sequence is 450bps. Also we consider the support for an average pattern to be k (in reality there will be some patterns some support more than k. But for the purposes of this calculation we ignore them). If $k = n$, each pattern one instance on each sequence, so the coverage for five patterns is $\sim 100/450 \sim 20\%$. For arbitrary $k < n$ the number of patterns can be estimated as $5 \times n/k$.

A.1.3 Clustering Threshold

We wish to decide a cut-off for the number of pairs of nodes to be connected by an edge based on their sequence similarity scores. If we assume that the number of motifs is 1, we would need about $n(n-1)/2$ edges. However, in general there could be more number of motifs per sequence, and the edges forming the motif need not be the topmost edges. A factor 'c' is thus introduced to account for this affect. Typically c is set at 2. However, depending on the number of nodes in the graph, this constant can be increased to 4 to allow for possibility of existence of more motifs. Let z be the number of nodes in the graph. In our algorithm we set $c = 2$ for $z/n < 2$ and $c = 4$ for $z/n > 2$, arbitrarily. In most cases discussed in the results section, such an approach works.

A.2 IUPAC Nomenclature for Nucleotides

Symbol	Meaning	Origin of designation
G	G	Guanine
A	A	Adenine
T	T	Thymine
C	C	Cytosine
R	G or A	puRine
Y	T or C	pYrimidine
M	A or C	aMino
K	G or T	Keto
S	G or C	Strong interaction (3 H bonds)
W	A or T	Weak interaction (2 H bonds)
H	A or C or T	not-G, H follows G in the alphabet
B	G or T or C	not-A, B follows A
V	G or C or A	not-T (not-U), V follows U
D	G or A or T	not-C, D follows C
N	G or A or T or C	aNy

Table A.2: Summary of single-letter code recommendations [9]

Symbol	A	B	C	D	G	H	K	M	S	T	V	W	N
Complement	T	V	G	H	C	D	M	K	S	A	B	W	N

Table A.3: Definition of complementary symbols [9]

A.3 Code-check Experiment

A code-check test was performed on a set of random, unrelated sequences containing an artificially implanted motif. The idea of the experiment was to demonstrate that all parts of the code work as intended.

Experiment: 10 random sequences from a collection of 4405 upstream regions from E.coli were taken. Various instances of the motif `GATCGNNNNCGATC` of length 14 bps were planted into the 10 regions where the middle spacers were generated as random stretch of nucleotides. The sequences were trimmed to 450bp each, and inputted into TABS program. The parameters used, and the output produced at each stage are listed below.

Conclusion: The implanted motif was found by the algorithm successfully with a 100% sensitivity. Some other sites highly resembling the motif that existed in the random sequences were also found. The experiment demonstrated that the code works intended and that there are no coding errors.

1. Stage 1: Teiresias.All $l = 6, w = 20, k = 7$ were found. This generated 720 patterns.

2. Stage 2: Statistical Filtering: Top 12 patterns were selected on the basis of z-score. These patterns were:

Pattern No.	z-score	Pattern
Pattern 1	173.30	GATCG....CGATC
Pattern 2	79.37	GWTCG....MGRTC
Pattern 3	73.97	GATCG....CGAT
Pattern 4	68.91	GMTYG....CKATC
Pattern 5	68.50	ABBTAGSVHYD.TDDG
Pattern 6	67.83	GRTMG....YGATC
Pattern 7	67.03	AB.VHGRTCS.DWVYBAT
Pattern 8	66.32	GATYS....CGRTC
Pattern 9	66.12	GATYG....CGWYC
Pattern 10	65.95	GWTCR....CGMTC
Pattern 11	65.54	GMWYSB.B.CGRTCG
Pattern 12	63.47	GMWCS....CGATC

10 of these patterns match the description of the motif, and the first one matches it exactly with the highest z-score. The feature map is shown in figure A-1.

3. Stage 3: Convolution Phase: 5 motifs were found. The first one had a P value of $1e-186$ and had a 100% sensitivity and 71% specificity. The logo is shown in figure A-2. All 10 implanted sites were found, and additionally 4 others that looked similar were found:

GGTAGAACCTGATC
GATTCTAGCCGGTC
GCTTGGAAACTATC
GATTGTAGGCGTCC
GATCG....CGATC

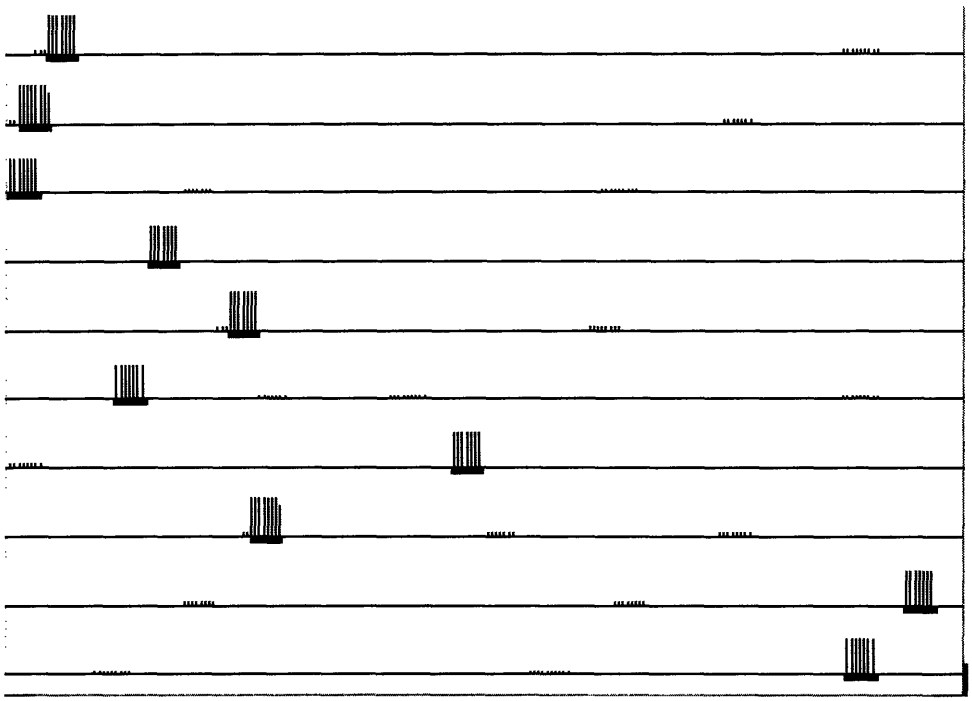


Figure A-1: Feature map showing the position of predicted patterns compared with the position of implanted motif (shown as a shaded bar)

Figure A-2: Sequence Logo corresponding to the best obtained motif, showing high degree of similarity with the implanted motif `GATCG....CGATC`.

A.4 Reported sites in RegulonDB

DNA-protein regulatory interaction reported in RegulonDB are based on validations using one or more of the following 6 methods:

1. 0. Computational Prediction
2. 1. Mutational analysis
3. 2. DNA footprinting
4. 3. Specific binding with crude extracts
5. 4. Consensus-based
6. 5. Changes in gene expression

0 and 4 are based on computational predictions (using sequence similarity), while 1,2 and 3 involve some kind of wet-laboratory experiments. 5 is based on changes in gene expression and hence does not necessary imply direct regulation. Except 5, there is a reported binding site corresponding to each regulatory interaction.

We can rank the reliability of the reported interactions in RegulonDB on the basis of the degree of confidence in the method used. 1,2 and 3 can be ranked first since they are identified by real experiments. 0 and 4 are both based on computational predictions. Method 0 involves the use of Patser (a weight-matrix-based search program that finds new candidate sites match the description of a weight-matrix constructed from an alignment of known sites [27]). 4 is based on sequence alignment between the "identified consensus" of a set of experimentally proven binding sites, and a new candidate site. If they share a high level of sequence similarity (no quantitative description of the level of similarity was found on RegulonDB website), the similarity is drawn. 0 and 4 are both ranked lower than 1,2 and 3 because they involve no

biological experiments. 5 is ranked the lowest, since the change in expression could just be due to indirect regulation.

For the 402 sites corresponding to the 30 transcription factors considered in this study, most sites were reported based on validation through at least 2 of the 6 methods. Moreover, no site was based on method 0. No sites were based on 5 *alone*. There were 96 sites that were reported based on 4 only (85), or both 4 and 5. considered H

A.5 Aligning Binding Sites

Creating alignments of binding sites, and building motifs can be a tricky issue because of various factors such as the sample size, the metric used for creating alignments, etc.

While any of several available multiple sequence alignment tools (e.g. ClustalW [43]) could be used for this purpose, Consensus is the most appropriate tool because of two reasons. First, Consensus works by maximizing the information content to find the best alignment which is the most popular metric for representing sites [37]. Second, Consensus scans several different possible lengths of the alignment and selects the best one, unlike ClustalW which only does global alignment and could miss conserved substrings.

The other issue is the directionality of the sequences used in building the alignments. While sites are reported in the literature from the 5'-3' end, there is no reason to believe that is the best orientation for aligning them. Protein molecules recognize sites in double-helical DNA structure which means there is no sense of directionality involved in the DNA-protein binding process.

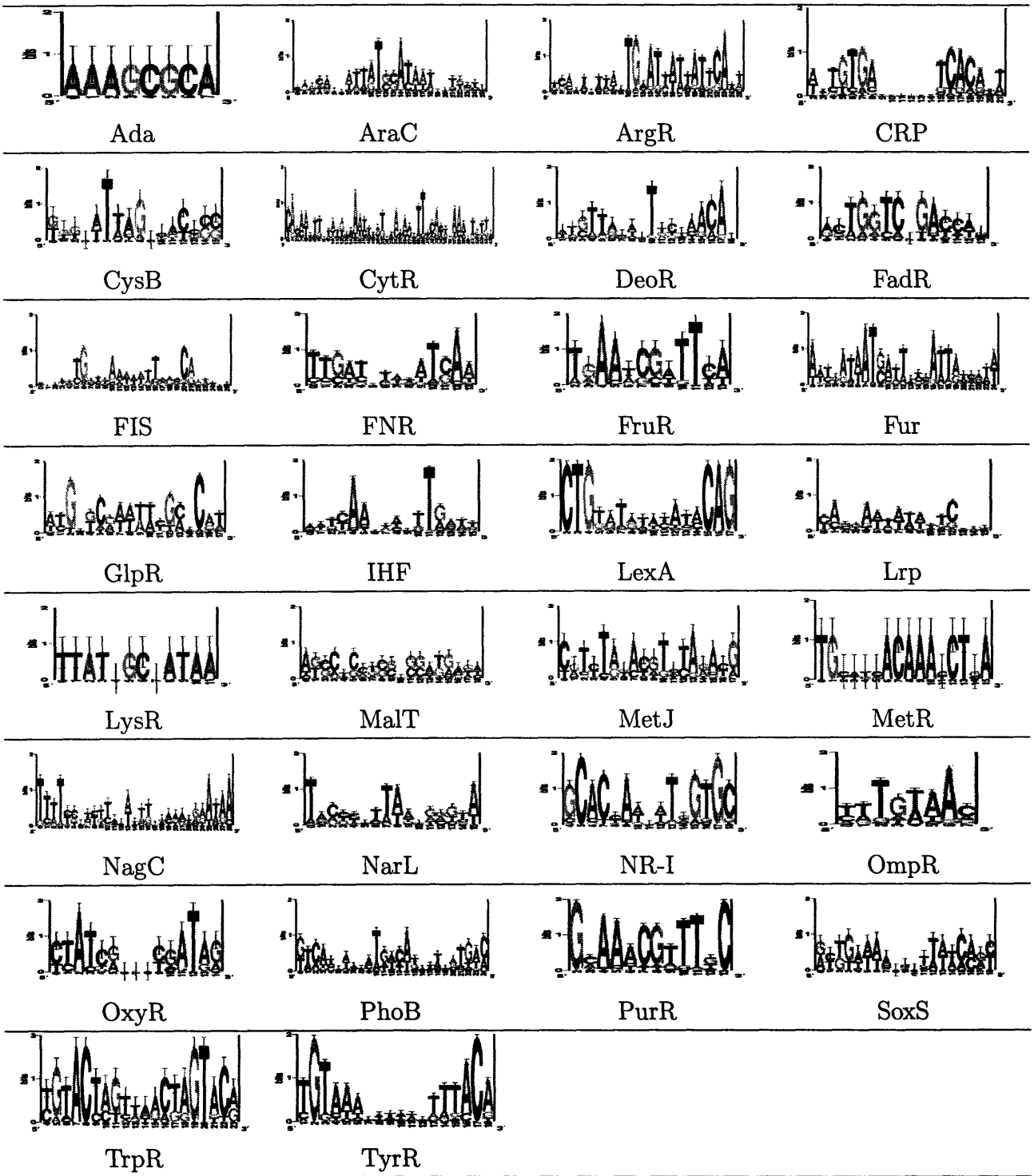


Figure A-3: Motifs corresponding to alignment of known sites for each regulon. The alignments were made from known sites reported in RegulonDB using Consensus program. Complementary strands were included in making the alignment, except for Ada and MetR.

From the database of known sites for the 30 E.coli regulons listed in section 4.1, we aligned sites bound by the same transcription factor using Consensus in two modes: (a) by including reverse strand sequences and (b) only with 5'–3' single strands sequences as reported. We found that mode (a) was able to identify far more interesting logos than (b). The reason was that by including the reverse strand sequences, signal is enhanced in sequences with palindromic motifs and a substantial number of binding motifs in E.coli happen to be palindromic (see figure A-3).

The alignments constructed were sent as input to Schneider's *makelogo* program for making graphic sequence logos. Unless otherwise mentioned, all alignments and motifs found are constructed by including reverse strand sequences.

A.6 Results Tables

Tables A.6 show the results from TABS up to filter 1 stage of the algorithm. The tables provide values for parameters used in each case, and also the intermediary figures, such as the number of patterns generated by Teiresias, the number of patterns selected after statistical filtering (column), minimum z-scores corresponding to the patterns selected, and the coverage (see section 3.3 for definition) of these patterns. In all these runs, l and w were kept at 6 and 20, respectively. The k values are reported. Sensitivity and specificity are reported based on two metrics - by *touch*, and by *fraction of overlapping bps*. Also reported is the mean width of *islands* generated by mapping the selected patterns on the feature map.

Table A.6 shows the detailed results for the full algorithm. It includes figures from the convolution phase. The total number of motifs found at the end of convolution, the size of the best motif, the number of input sequences in which the motif has an occurrence, the width of the motif, its P value and the sensitivity and specificity of the best motif. In the final row of the table, the mean values for every column are calculated.

Tables A.6, A.7 and A.8 show the results from the other algorithms, namely Consensus and AlignACE.

Table A.4: Detailed results at Filter 1 Stage

Regulon	Size	Teiresias		Statistical Filtering							Mean Width of Islands, bps
							Touch		Fraction of bps overlap		
		k	No. of Patterns	No. of Patterns	z-score	Coverage	Sensitivity	Specificity	Sensitivity	Specificity	
Ada	3	3	27601	5	1256405.62	0.33	0.5	0.08	0.05	0.01	46
AraC	5	4	38058	6	6314.72	0.28	0.67	0.42	0.46	0.3	39
ArgR	7	6	5008	6	641.16	0.12	0.83	0.71	0.52	0.46	35
CRP	73	22	132660	17	15.06	0.04	0.52	0.52	0.2	0.45	15
CysB	5	4	44723	6	8682.77	0.25	1	0.43	0.6	0.26	38
CytR	7	6	3164	6	123.09	0.23	0.88	0.39	0.51	0.27	29
DeoR	3	3	27512	5	1145427.85	0.38	0.5	0.13	0.55	0.14	47
FadR	4	4	4920	5	3656.87	0.27	0.5	0.25	0.15	0.06	36
FIS	25	9	110744	14	939.64	0.07	0.03	0.02	0	0	30
FNR	21	9	175101	12	48.01	0.18	0.67	0.18	0.4	0.17	30
FruR	7	6	5436	6	127.9	0.22	1	0.17	0.62	0.12	29
Fur	10	7	15590	7	710.78	0.09	0.67	0.2	0.44	0.25	35
GlpR	4	4	12211	5	9803.12	0.22	0.8	0.82	0.59	0.69	34
IHF	22	9	269272	12	56.78	0.21	0.76	0.24	0.49	0.13	30
LexA	9	6	9040	8	225.06	0.11	0.89	0.7	0.64	0.5	28
Lrp	14	8	62625	9	60.88	0.2	0.69	0.26	0.43	0.16	30
LysR	3	3	34811	5	1973116.01	0.33	0	0	0	0	50
MalT	4	4	8211	5	3700.8	0.26	0.56	0.24	0.28	0.16	36
MetJ	3	3	32124	5	2173186.12	0.29	1	0.38	0.77	0.2	45
MetR	3	3	28034	5	792420.78	0.35	1	0.2	0.94	0.22	50
NagC	4	4	15728	5	5581.59	0.23	0.5	0.29	0.33	0.21	35
NarL	12	7	36948	9	83.11	0.23	0.62	0.17	0.42	0.15	27
NR-I	3	3	26874	5	587872.29	0.32	0.78	0.33	0.68	0.33	49
OmpR	6	5	56737	6	824.71	0.25	0.79	0.34	0.49	0.24	33
OxyR	4	4	12611	5	7057.21	0.18	0.75	0.27	0.18	0.14	39
PhoB	5	4	39310	6	20632.06	0.17	0.71	0.53	0.66	0.39	40
PurR	17	8	32061	11	114.1	0.08	0.85	0.61	0.6	0.49	27
SoxS	5	4	26730	6	2606.62	0.29	0.25	0.04	0.13	0.02	33
TrpR	5	4	24872	6	7089.95	0.33	0.8	0.29	0.71	0.23	36
TyrR	8	6	4497	7	215.69	0.07	0.73	0.9	0.44	0.9	23

Regulon	Size	Teiresias		Statistical Filtering			Convolution							
		k	No. of Patterns	z-score	Fraction of bps overlap		No. of motifs	Top motif						
					Sensitivity	Specificity		Sites	Genes	Width	P value	Sensitivity	Specificity	Corrected Specificity
Ada	3	3	27601	1256405.62	0.05	0.01	3	3	3	20	-54.83	0	0	0
AraC	5	4	38058	6314.72	0.46	0.3	5	4	4	25	-101.65	0.25	0.75	0.75
ArgR	7	6	5008	641.16	0.52	0.46	6	7	6	27	-209.55	0.83	0.71	0.99
CRP	73	22	132660	15.06	0.2	0.45	1	93	57	13	-391.68	0.49	0.45	0.54
CysB	5	4	44723	8682.77	0.6	0.26	5	5	4	21	-104.79	0.5	0.4	0.50
CytR	7	6	3164	123.09	0.51	0.27	1	5	5	15	-64.79	0	0	0
DeoR	3	3	27512	1145427.85	0.55	0.14	6	4	3	18	-67.11	0.5	0.5	0.67
FadR	4	4	4920	3656.87	0.15	0.06	1	5	4	7	-14.66	0.17	0.2	0.2
FIS	25	9	110744	939.64	0	0	1	26	19	16	-144.6	0.03	0.04	0.04
FNR	21	9	175101	48.01	0.4	0.17	11	11	10	20	-176.28	0.06	0.09	0.12
FruR[a]	7	6	5436	127.9	0.62	0.12	5	21	7	12	-45.11	0.75	0.6	1
Fur	10	7	15590	710.78	0.44	0.25	2	12	10	29	-338.37	0.67	0.25	0.6
GlpR	4	4	12211	9803.12	0.59	0.69	1	4	4	22	-87.94	0.4	1	1
IHF	22	9	269272	56.78	0.49	0.13	0	-	-	-	-	0	0	0
LexA	9	6	9040	225.06	0.64	0.5	7	8	7	16	-120.77	0.78	0.88	1
Lrp	14	8	62625	60.88	0.43	0.16	21	13	10	18	-93.04	0.19	0.23	0.37
LysR	3	3	34811	1973116.01	0	0	5	4	3	28	-118.2	0	0	0
MalT	4	4	8211	3700.8	0.28	0.16	1	4	4	4	4.02	0.44	0.75	0.75
MetJ	3	3	32124	2173186.12	0.77	0.2	3	3	3	22	-62.15	1	0.67	1
MetR	3	3	28034	792420.78	0.94	0.22	2	3	3	6	-3.62	0.67	0.67	1
NagC	4	4	15728	5581.59	0.33	0.21	1	4	4	19	-72.61	0.17	0.2	0.2
NarL	12	7	36948	83.11	0.42	0.15	-	-	-	-	-	0	0	0
NR-I	3	3	26874	587872.29	0.68	0.33	1	3	3	15	-36.54	0.56	1	1
OmpR	6	5	56737	824.71	0.49	0.24	8	5	4	20	-97.04	0.29	0.4	0.67
OxyR	4	4	12611	7057.21	0.18	0.14	2	5	4	17	-80.25	0.25	0.2	0.2
PhoB	5	4	39310	20632.06	0.66	0.39	6	6	4	27	-177	0.71	0.83	0.83
PurR	17	8	32061	114.1	0.6	0.49	1	21	15	12	-218.01	0.85	0.52	0.57
SoxS	5	4	26730	2606.62	0.13	0.02	1	4	4	20	-76.11	0	0	0
TrpR[a]	5	4	24872	7089.95	0.71	0.23	3	4	4	16	-55.67	0.8	1	1
TyrR	8	6	4497	215.69	0.44	0.9	1	11	8	16	-173.22	0.73	0.91	0.91
Avg.	10.03	5.73	44107.1	266924.68	0.44	0.26	3.83	10.64	7.71	17.89	-113.63	0.4	0.44	0.53

[a]2nd best motif

Table A.5: Complete Results for the 30 E.coli Regulons

Regulon	Size	Sites	Genes	Width, bps	ln(P value)	Sensitivity	Specificity	Corrected Specificity [a]
Ada	3	3	3	14	-32.88	0	0	0
AraC[b]	5	5	5	137	-302.29	0.75	1	1
ArgR	7	7	7	39	-326.17	0.83	0.71	0.83
CRP	73	73	73	75	-646.35	0.47	0.51	0.7
CysB	5	5	5	16	-74.39	0	0	0
CytR	7	8	8	22	-180.76	0.78	0.88	0.88
DeoR	3	3	3	25	-73.13	0.25	0.33	0.5
FadR	4	4	4	8	-16.41	0.17	0.25	0.25
FIS	25	25	25	52	-505.25	0	0	0
FNR	21	21	21	23	-107.95	0	0	0
FruR	7	7	7	16	-108.58	1	0.57	1
Fur	10	10	10	43	-354.18	0.67	0.3	0.75
GlpR[b]	4	4	4	91	-331.19	0.67	1	1
IHF[b]	22	22	22	89	-271.1	0.71	0.41	0.82
LexA	9	9	9	20	-192.21	0.89	0.78	0.88
Lrp	14	14	14	6	-32.11	0.07	0.07	0.12
LysR[b]	3	3	3	172	-556.33	1	0.33	0.99
MalT	4	4	4	8	-16.41	0.56	0.75	0.75
MetJ	3	3	3	22	-62.15	0.6	0.67	1.01
MetR	3	3	3	11	-21.91	0	0	0
NagC[b]	4	5	5	174	-350.01	0.71	0.6	0.6
NarL	12	12	12	14	-160.63	0.17	0.17	0.29
NR-I	3	3	3	15	-36.54	0.44	1	1
OmpR	6	6	6	10	-43.42	0	0	0
OxyR	4	4	4	13	-41.96	0	0	0
PhoB	5	5	5	16	-74.39	0	0	0
PurR	17	17	17	16	-279.53	0.92	0.71	0.93
SoxS	5	5	5	9	-28.48	0	0	0
TrpR	5	5	5	20	-100.63	1	1	1
TyrR	8	8	8	18	-147.49	0.47	0.88	0.88

[a]Calculated by excluding predicted sites on genes for which there is no reported site
[b]Width of predicted motif too long

Table A.6: Consensus-double Results

Regulon	Size	Sites	Genes	Width, bps	ln(P value)	Sensitivity	Specificity	Corrected Specificity [a]
Ada	3	3	3	14	-32.88	0	0	0
AraC	5	5	5	7	-15.35	0	0	0
ArgR	7	7	7	37	-307.25	0.83	0.71	0.83
CRP	73	73	73	34	-71.79	0.59	0.57	0.79
CysB	5	5	5	20	-100.62	0	0	0
CytR	7	8	8	6	-16.57	0	0	0
DeoR	3	3	3	9	-14.59	0.25	0.33	0.5
FadR	4	4	4	6	-6.2	0	0	0
FIS	25	25	25	49	-579.7	0.04	0.04	0.04
FNR	21	21	21	21	-315.42	0.44	0.38	0.53
FruR	7	7	7	21	-155.88	0.5	0.29	0.51
Fur	10	10	10	28	-325.61	0.78	0.4	1
GlpR	4	4	4	24	-98.16	0.33	1	1
IHF	22	22	22	15	-207.56	0	0	0
LexA	9	9	9	20	-192.21	0.89	0.78	0.88
Lrp	14	14	14	10	-110.57	0	0	0
LysR	3	3	3	23	-65.8	0	0	0
MalT	4	4	4	21	-82.84	0.22	0.5	0.5
MetJ	3	3	3	20	-54.83	1	0.67	1.01
MetR	3	3	3	16	-40.2	0	0	0
NagC	4	5	5	11	-41.59	0	0	0
NarL	12	12	12	14	-160.63	0.08	0.08	0.14
NR-I	3	3	3	16	-40.2	0.44	1	1
OmpR	6	6	6	31	-187.32	0	0	0
OxyR	4	4	4	12	-36.86	0	0	0
PhoB	5	5	5	7	-15.35	0	0	0
PurR	17	17	17	17	-303.5	0.92	0.71	0.93
SoxS	5	5	5	13	-54.71	0	0	0
TrpR	5	5	5	22	-113.74	1	1	1
TyrR	8	8	8	19	-158.4	0.47	0.88	0.88

[a]Calculated by excluding predicted sites on genes for which there is no reported site

Table A.7: Consensus-single Results

Regulon	Size	Sites	Genes	Width, bps	ln(P value)	Sensitivity	Specificity	Corrected Specificity [a]
Ada	3	26	3	23	-206.7	1	0.08	0.12
AraC	5	40	5	18	-240.85	0.75	0.28	0.28
ArgR	7	54	7	15	-255.31	0.25	0.06	0.07
CRP	73	578	72	10	nan	0.32	0.05	0.07
CysB	5	37	5	15	-147.14	1	0.14	0.18
CytR	7	61	8	15	-213.19	0	0	0
DeoR	3	32	3	14	-167.62	1	0.16	0.24
FadR	4	30	4	23	-162.34	1	0.17	0.17
FIS	25	287	25	10	nan	0.54	0.05	0.05
FNR	21	145	21	18	nan	0.44	0.06	0.08
FruR	7	48	7	18	-194.62	0.5	0.04	0.07
Fur	10	110	10	16	-326.26	0.22	0.02	0.05
GlpR	4	34	4	20	-211.11	0.47	0.18	0.18
IHF	22	157	22	19	nan	0.18	0.02	0.04
LexA	9	81	9	19	-269.57	0.78	0.09	0.1
Lrp	14	109	14	16	-224.94	0.21	0.03	0.05
LysR	3	17	3	27	-172.5	0	0	0
MalT	4	-	-	-	-	-	-	-
MetJ	3	25	3	14	-190.73	0.2	0.04	0.06
MetR	3	21	3	23	-141.89	0.33	0.05	0.08
NagC	4	39	5	20	-208.56	0.29	0.05	0.05
NarL	12	92	12	17	-214.35	0.33	0.02	0.03
NR-I	3	13	3	18	-226.19	0.89	0.54	0.54
OmpR	6	30	6	13	-162.37	0	0	0
OxyR	4	21	4	25	-122.48	0.75	0.14	0.14
PhoB	5	59	5	16	-213.14	0.6	0.07	0.12
PurR	17	194	17	16	nan	0.92	0.09	0.12
SoxS	5	49	5	16	-203.76	0.5	0.04	0.07
TrpR	5	51	5	22	-159.82	0.6	0.06	0.06
TyrR	8	69	8	20	-224.47	0.6	0.13	0.13

[a]Calculated by excluding predicted sites on genes for which there is no reported site

Table A.8: AlignACE Results

Bibliography

[1] Timothy L. Bailey and Charles Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology*, pages 28–36, 1994.

[2] Esperanza Benítez-Bellón, Gabriel Moreno-Hagelsieb, and Julio Collado-Vides. Evaluation of thresholds for the detection of binding sites for regulatory proteins in escherichia coli k12 dna. *Genome Biology*, 3:1–16, 2002.

[3] Alvis Brazma, Inge Jonassen, Jaak Vilo, and Esko Ukkonen. Predicting gene regulatory elements in silico on a genomic scale. *Genome Research*, 8:1202–1215, 1998.

[4] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[5] TA Brown. *Genomes 2.* Wiley-Liss, 2002.

[6] C. Burge and S. Karlin. Prediction of complete gene structures in human genomic dna. *Journal of Molecular Biology*, 268(1):78+, 1997.

[7] Harmen J. Bussemaker, Hao Li, and Eric D. Siggia. From the cover: Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis. *PNAS*, 97:10096–10100, 2000.

[8] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms.* McGraw Hill, 1990.

[9] A Cornish-Bowden. Nomenclature for incompletely specified bases in nucleic acid sequences: recommendations 1984. *Biochem. J.*, 229:281–286, 1985.

[10] Alain Denise, Mireille Rgnier, and Mathias Vandenbogaert. Assessing the statistical significance of overrepresented oligonucleotides. *Algorithms in Bioinformatics: First International Workshop, WABI 2001, Aarhus, Denmark, August 28-31, 2001, Proceedings*, 2149:85–97, 2001.

[11] Aristidis Floratos. *Pattern Discovery in Biology: Theory and Applications.* PhD thesis, Department of Computer Science, New York University, 1999.

[12] Jacques van Helden, B. Andr, and J. Collado-Vides. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *Journal of Molecular Biology*, 281:827–842, 1998.

[13] Jacques van Helden, Alma. F. Rios, and Julio Collado-Vides. Discovering regulatory elements in non-coding sequences by analysis of spaced dyads. *Nucl. Acids. Res.*, 28:1808–1818, 2000.

[14] GZ Hertz and GD Stormo. Identifying dna and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15:563–577, 1999.

[15] R.A. Howard. *Dynamic Probabilitic Systems Vol. I: Markov Models.* John Wiley & Sons, 1971.

[16] L.C.K Hui. Color set size problem with applications to string matching. *Combinatorial Pattern Matching*, 644:230–243, 1992.

[17] Tien Huynh, Isidore Rigoutsos, Laxmi Parida, Daniel Platt, and Tetsuo Shibuya. The web server of ibm's bioinformatics and pattern discovery group. *Nucl. Acids. Res.*, 31:3645–3650, 2003. http://cbcsrv.watson.ibm.com/Tspd.html.

[18] J. Kleffe and M. Borodovsky. First and second moments of counts of words in random texts generated by markov chains. *Computer Applications in the Biosciences*, 8:433+, 1992.

[19] Charles E. Lawrence, Stephen F. Altschul, Mark S. Boguski, Jun S. Liu, Andrew F. Neuwald, and John C. Wootton. Detecting subtle sequence signals: a gibbs sampling strategy for multiple alignment. *Science*, 262:208–214, 1993.

[20] Ming-Ying Leung, Genevieve M. Marsh, and Terence P. Speed. Over- and underrepresentation of short dna words in herpesvirus genomes. *Journal of Computational Biology*, 3:345–360, 1996.

[21] Hao Li, Virgil Rhodius, Carol Gross, and Eric D. Siggia. Identification of the binding sites of regulatory proteins in bacterial genomes. *PNAS*, 99:11772–11777, 2002.

[22] A.V. Lukashin and M. Borodovsky. Genemark.hmm: new solutions for gene finding. *Nucleic Acids Research*, 26:1107+, 1998.

[23] http://prodes.toulouse.inra.fr/prodom/current/html/home.php.

[24] Zhaohui S. Qin1, Lee Ann McCue, William Thompson, Linda Mayerhofer, Charles E. Lawrence, and Jun S. Liu. Identification of co-regulated genes through bayesian clustering of predicted regulatory binding sites. *Nature Biotechnology*, 21:435–439, 2003.

[25] K Quandt, K Frech, H Karas, E Wingender, and T Werner. Matind and matinspector: new fast and versatile tools for detection of consensus matches in nucleotide sequence data. *Nucl. Acids Res.*, 23:4878–84, 1995.

[26] M. Régnier and W. Szpankowski. On pattern frequency occurrences in a markovian sequences. *Algorithmica*, 22:631–649, 1998.

[27] http://www.cifn.unam.mx/Computational-Genomics/regulondb/.

[28] I Rigoutsos and A Floratos. Combinatorial pattern discovery in biological sequences: The teiresias algorithm. *Bioinformatics*, 14:55–67, 1998. [published erratum appears in Bioinformatics 1998;14(2):229].

[29] Isidore Rigoutsos, Aris Floratos, Christos Ouzounis, Yuan Gao, and Laxmi Parida. Dictionary building via unsupervised hierarchical motif discovery in the sequence space of natural proteins. *PROTEINS: Structure, Function, and Genetics*, 37:264–277, 1999.

[30] Isidore Rigoutsos, Aris Floratos, Laxmi Parida, Yuan Gao, and Daniel Platt. The emergence of pattern discovery techniques in computational biology. *Metabolic Engineering*, 159:159–177, 2000.

[31] Isidore Rigoutsos, Tien Huynh, Aris Floratos, Laxmi Parida, and Daniel Platt. Dictionary-driven protein annotation. *Nucl. Acids. Res.*, 30:3901–3916, 2002.

[32] K Robinson, AM McGuire, and George M. Church. A comprehensive library of dna-binding site matrices for 55 proteins applied to the complete escherichia coli k12 genome. *Journal of Molecular Biology*, 284:241–254, 1998.

[33] Frederick P. Roth, Jason D. Hughes, and George M. Church. Finding dna regulatory motifs within unaligned noncoding sequences clustered by whole-genome mrna quantification. *Nature Biotechnology*, 16:939–945, 1998.

[34] Heladia Salgado, Alberto Santos-Zavaleta, Socorro Gama-Castro, Dulce Millan-Zarate, Edgar Diaz-Peredo, Fabiola Sanchez-Solano, Ernesto Perez-Rueda, Cesar Bonavides-Martinez, and Julio Collado-Vides. Regulondb (version 3.2): transcriptional regulation and operon organization in escherichia coli k-12. *Nucl. Acids. Res.*, 29:72–74, 2001.

[35] M Schena, D Shalon, RW Davis, and PO Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270:467–470, 1995.

[36] T. D. Schneider. Information content of individual genetic sequences. *J. Theor. Biol.*, 189(4):427–441, 1997.

[37] T. D. Schneider and R. M. Stephens. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.*, 18:6097–6100, 1990.

[38] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication.* University of Illinois Press, 1963.

[39] Tetsuo Shibuya and Isidore Rigoutsos. Dictionary-driven prokaryotic gene finding. *Nucl. Acids. Res.*, 30:2701–2725, 2002.

[40] Saurabh Sinha. Composite motifs in promoter regions of genes: models and algorithms. Technical report, University of Washington, 2001.

[41] Saurabh Sinha and Martin Tompa. Discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Research*, 30(24):5549+, 2002.

[42] Gary D. Stormo and Dana S. Fields. Specificity, free energy and information content in protein-dna interactions. *TIBS*, 23:109–113, 1998.

[43] JD Thompson, DG Higgins, and TJ Gibson. Clustalw: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl. Acids. Res.*, 22:4673–4680, 1994.

[44] Robert OJ Weinzierl. *Mechanisms of Gene Expression.* Imperial College Press, 1999.

[45] E. Wingender, X. Chen, R. Hehl, H. Karas, V. Liebich, I.and Matys, T. Meinhardt, M. Pr, I. Reuter, and F. Schacherer. Transfac: an integrated system for gene expression regulation. *Nucleic Acids Res.*, 28:316–319, 2000.

[46] R. F. Yeh, Fairbrother, Sharp W., P. A., and C. B. Burge. Predictive identification of exonic splicing enhancers in human genes. *Science*, 297:1007+, 2002.

[47] J Zhu and MQ Zhang. Scpd: a promoter database of the yeast saccharomyces cerevisiae. *Bioinformatics*, 15:607–611, 1999.

Appendix A.7

Capstone Report
Commercialization of Bioinformatics Tools

by
Vipin Gupta

Submitted: Aug 31, 2004

Table of Contents

List of Figures

List of Tables

1 Overview

This report addresses the business and market issues related to commercialization of bioinformatics tools. It covers an introduction to the bioinformatics industry, market segments, business models, the current market trends, the competitive landscape, as well as a future outlook. In the final section the report addresses the question of how one would go about commercializing the "Regulatory Sequence Analysis Package" produced as a result of the work in this thesis.

2 Introduction to the Bioinformatics Industry

Over the past two decades, the pharmaceutical industry has experienced a fundamental shift in drug discovery processes. Although the industry was once highly dependent on "shotgun-based" chemistry techniques, this serendipitous approach has slowly given way to the emerging field of genomics. On a fundamental basis, genomics reveals the basis for human disease at the molecular level. By understanding how life's processes are carried out in normal situations, scientists can identify how mutations at the genetic level can lead to illnesses. Furthermore, by linking specific genes with specific diseases, researchers can design targeted compounds to address cellular malfunctions.

Initially, one of the major obstacles to the genomic revolution was the general lack of biological information. To truly understand this emerging discipline requires massive amounts of gene sequence, gene expression, and proteomic data. Throughout the early 1990s, labor-intensive methods to secure this information fell short of expectations.

In response to an outcry for a more efficient and accurate data collection infrastructure, a new life sciences industry emerged in applying industrial-scale techniques to laboratory processes. These new tools (e.g., gene sequencers, microarrays, and mass spectrometers) and high-throughput techniques have unleashed a flood of new biological data - information that continues to double in size every 12 months.

The aforementioned events have shifted the bottleneck in drug discovery from data collection to data analysis, interpretation, and integration. This has spawned an exciting new discipline (informatics) that employs information technology to drive life science discoveries. Informatics is defined as software tools that permit a scientist to capture, visualize, and analyze life science related data. Specifically, informatics deals with database management, database integration, and complex mining algorithms that allow a researcher to harness relevant information from a multitude of assays or experiments.

2.1 Factors Driving Demand

Informatics tools allow for more rapid screening of genetic data. Informatics has become crucial for documenting, storing and analysing today's information-rich environment. Use of visualization tools and sophisticated algorithms allow scientists to identify complex relationships in large data sets. This leads to better productivity and, given today's drug discovery environment, informatics lends a distinct advantage.

In the big picture of things, this technology has the ability to reduce failures in clinical trials. For example, the average pharmaceutical company ties up 50% of its research and development dollars on clinical trials, and 90% of compounds that enter clinical trials never reach the market. By creating more effective drug compounds (through molecular modeling) and filtering out unpromising leads, pharmaceutical manufacturers could save millions of dollars by foregoing the development of drugs that are ultimately doomed for failure. As shown in Figure 1, the key is to "fail early and often" - maximizing the focus in drug discovery and minimizing the expense of clinical trials. In the remainder of this section we discuss the factors that are driving the demand for bioinformatics.

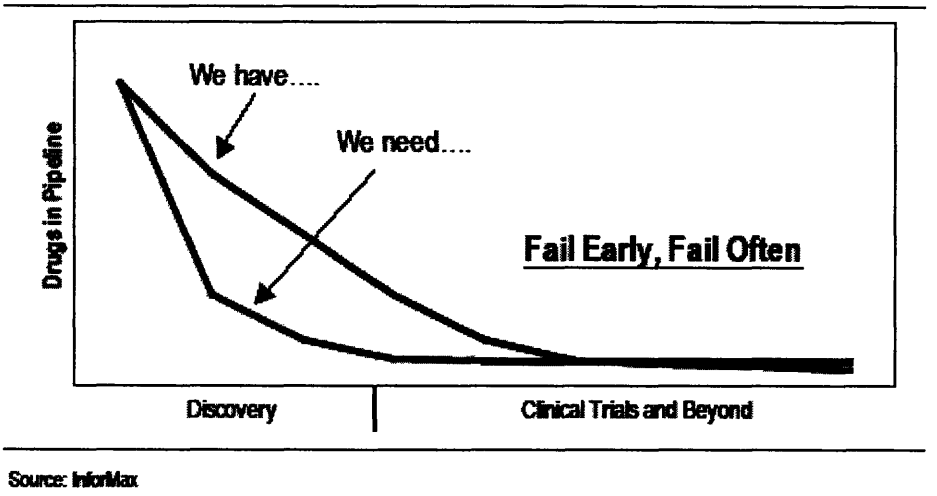


Figure 1: Enhancing Clinical Trials

2.1.1 Competitive Pressures in Big Pharmaceuticals

Over the past two decades, investors have witnessed tremendous growth in new pharmaceutical and biotechnology ventures, both in the United States and abroad. As a result, many large cap pharmaceutical companies have found themselves immersed in an intense competitive landscape. Although these global conglomerates once "sat on their laurels" with successful new drugs, today's environment requires full drug pipelines to sustain continued growth. Although the rules have changed, the pharmaceutical industry has struggled to answer the call. As shown in Table 1, the pipeline of new drug application (NDA) approvals has declined 45% since 1996, and generic approvals have increased 41% since 1999. This is due to the fact that discovering new drugs has become extremely difficult due to technology limitations (much of the "low-hanging fruits" already in the market). Also, the fact that many drug patents are set to expire in the foreseeable future, the will likely result in a flood of new generic drugs, which will erode the market share of established pharmaceutical companies. 90% of drugs that enter clinical trials ultimately fail to reach the market for a number of reasons (e.g. toxicity, adverse side effects, lack of efficacy, and so forth). New drug companies must increase productivity to maintain growth rates and reduce costs. Today's drug discovery process takes 12-15 years to bring a new drug to the market, with an average cost of $800bn.

Year	NDA Approvals	Generic Drug
1995	82	207
1996	131	212
1997	121	273
1998	90	225
1999	83	186
2000	98	244
2001	66	234
2002	78	321
2003	72	263

Source: CDER Report to the Nation 2003

Table 1: Number of NDA and Generic Drug Approvals by Year

2.1.2 In-House Solutions are Falling Short

With information technology presented as a potential enabler to an inefficient drug discovery process, many pharmaceutical companies have turned to informatics to accelerate drug discovery and make clinical trials more efficient. Initially, however, specialized third-party applications were largely unavailable and limited in functionality.

To overcome this hurdle, many organizations invested substantial resources to develop in-house productivity solutions. Unfortunately, given the dependence on off-the-shelf applications (e.g., Microsoft Excel) these solutions defied the concepts of user friendliness and required complex algorithmic programming. In addition, the scalability of these applications was eventually strained by a flood of genomic data wrought forth by industrial-scale equipment and techniques. Most important, to maintain these complex solutions, organizations require trained "informaticians," individuals who possess familiarity with computer science as well as genomics. As one might expect, finding employees with these qualifications was (and still is) a formidable task. The market for these individuals has also been strained by the evolution of new informatics ventures - start-ups that offer attractive pay packages and upside from options.

2.1.3 Data Analysis is the New Bottleneck

A third factor influencing the demand for informatics is attributable to the advent of industrial scale biology - with new data collection technologies (e.g., DNA synthesizers and microarrays) providing an onslaught of gene sequence, gene

expression, and proteomic data. Although the industry once suffered from a lack of qualified targets and candidate drugs, lead scientists must now decide where to start amidst the overload of biological data. This phenomenon has shifted the bottleneck in drug discovery from data collection to data analysis, interpretation, and integration.

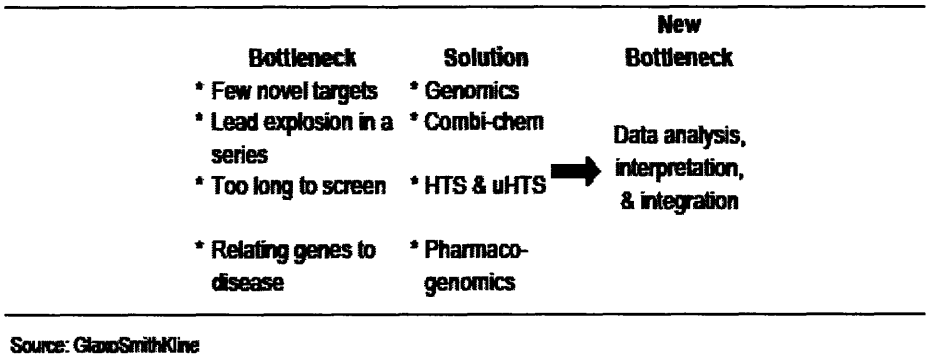
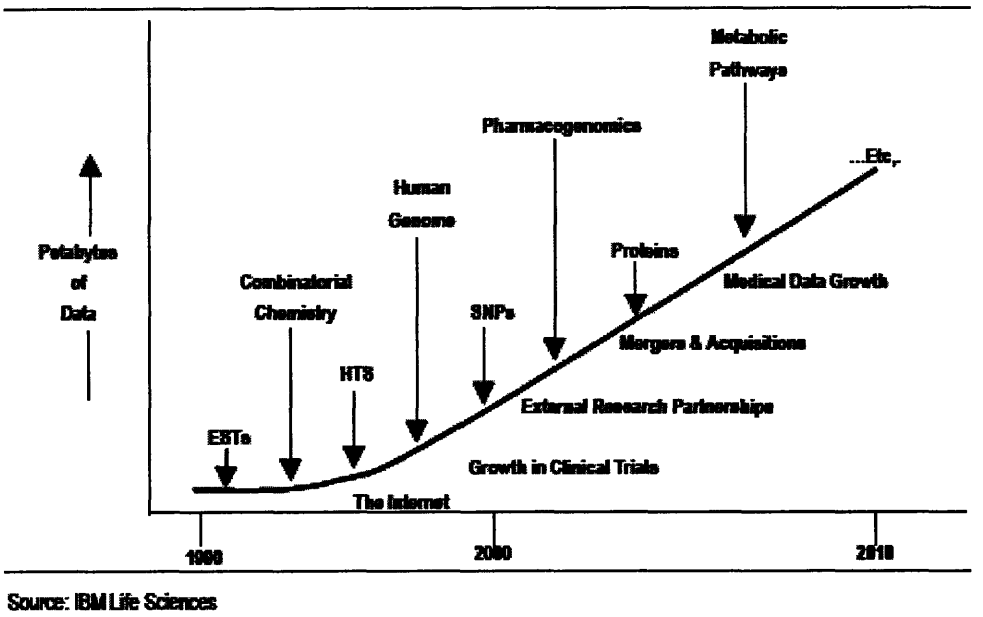


Figure 2: Drug Discovery Today

2.2 The Overload of Biological Data

The dramatic improvement in data collection methodologies has been viewed as a godsend by many, with more information fueling exponential advances in productivity. Using today's technology, scientists can run large-scale experiments in a matter of hours rather than weeks, and access to data is now considered a non-issue rather than a hurdle. These advances have opened the door to several new areas of science - such as combinatorial chemistry, high-throughput screening (HTS), and genomics (with a draft of the human genome now complete). See Figure 3. According to UBS Warburg, genomic data is expected to at least double every 12 months.



Source: IBM Life Sciences

Figure 3: Technologies Contributing to the Flood of Information

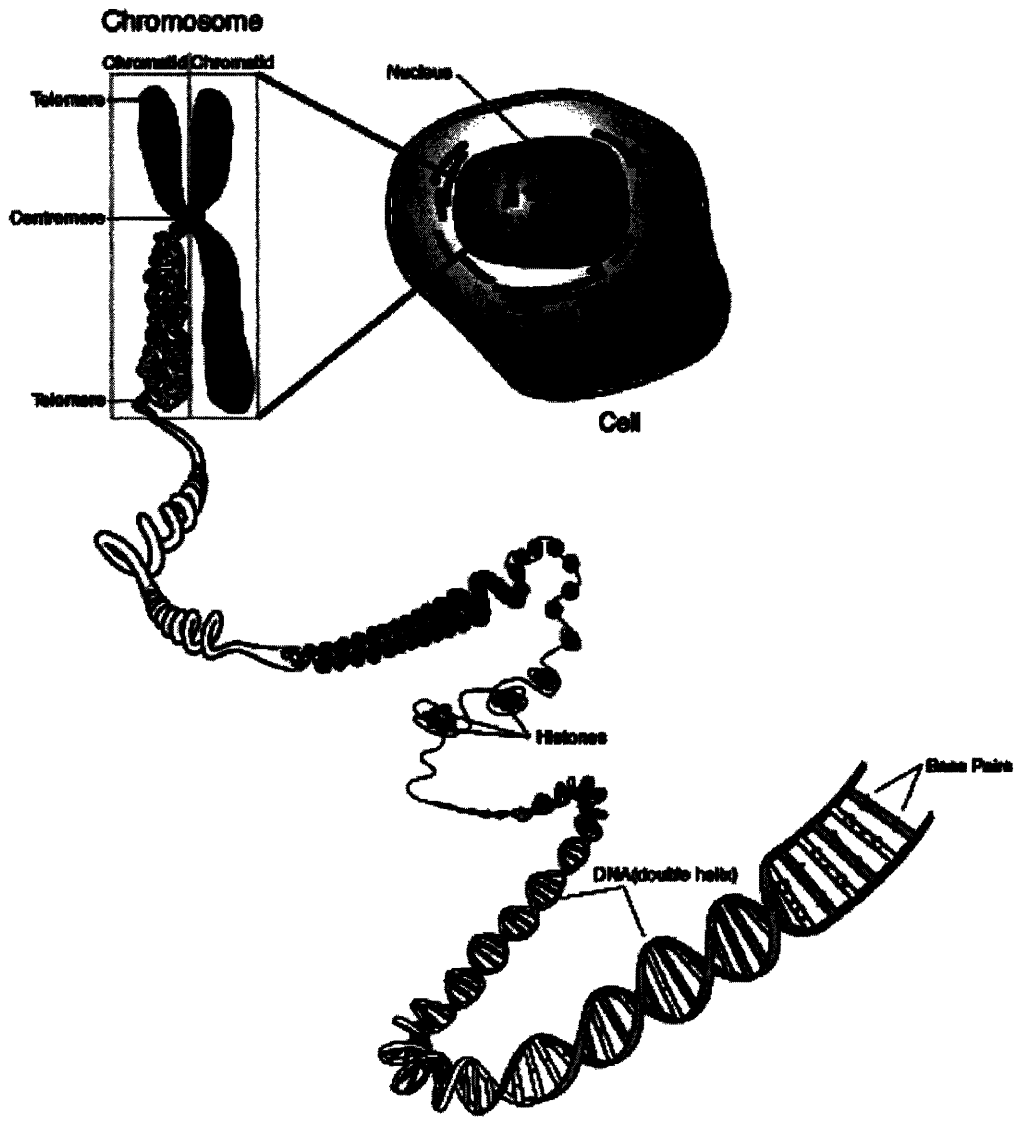
3 Areas of Application

As mentioned previously, competition in the informatics market is typically bifurcated by scientific discipline. We have therefore segmented our discussion into the various fields of: (1) gene sequencing; (2) comparative genomics; (3) functional genomics; (4) pharmacogenomics; (5) structural genomics; (6) other life science informatics categories. Hereafter, we describe the opportunity for informatics within the framework of these fields.

3.1 Gene Sequence Analysis

DNA represents the information warehouse present in every cell of every organism on the planet. The code for carrying out and maintaining life lies within a helical bundle of an intricately ordered array of four distinct chemical bases: A, C, T, and G. A and T can form base pairs, as can C and G, and the ordered sequence of these bases represents the blueprint for creating proteins that carry out all of life's processes. As shown in Figure 4, this double helix folds into a tight chromosome structure in the nucleus of a cell.

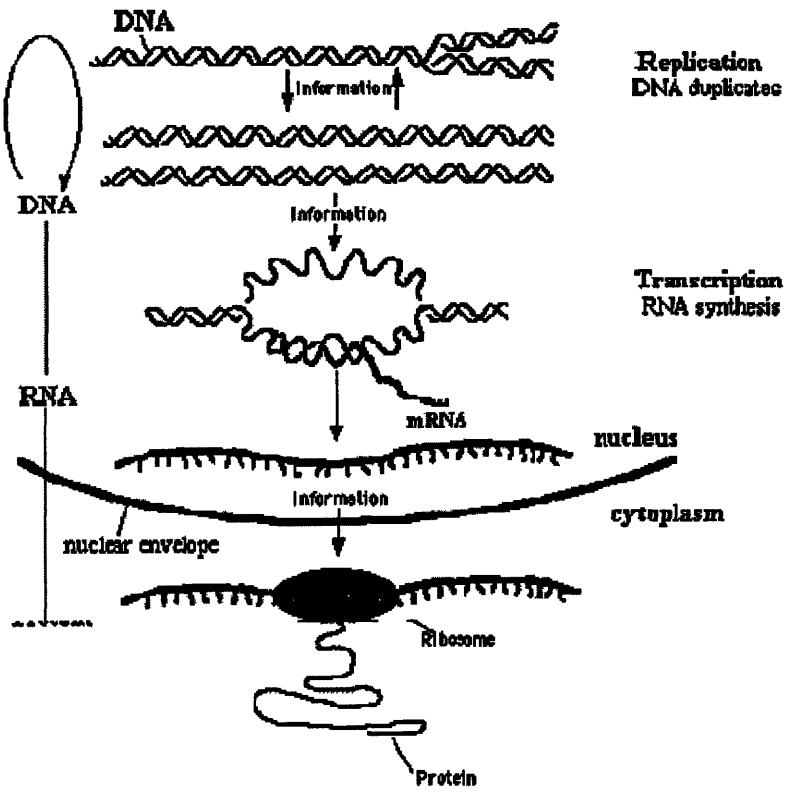
The entire DNA code for an organism is termed its genome, and it is generally housed in multiple chromosomes (in humans there are a total of 23 pairs of chromosomes). As one might expect, the length of each genome varies from species to species. For example, the genome of E. coli is 4.6 million base pairs, whereas the genome of humans is on the order of 3.0 billion base pairs. If put on paper, the genome of humans would fill about 200 1,000-page telephone books.



Source: Access Excellence

Figure 4: Structure of DNA

Throughout the genome of humans, it is believed that approximately 30,000 genes exist - with only around 10,000 discovered to date. A gene is simply defined as a region of the genome that can vary in size and complexity and codes for a specific protein. As shown in Chart 19, in order for the gene to perform its function, its sequence of bases must be read and copied by cellular machines into a single-stranded message called RNA. The purpose of the RNA molecule is to deliver DNA "instructions" to the protein synthesizing machinery of the cell. As such, for each gene to be expressed, or "turned on," it must first be made into a message before it is encoded into a protein. This pattern of events is ongoing and occurs each time that the cell makes a protein. The cycle of genes that is turned on and off in a cell represents the entire regulatory network for that cell at a given point in time.



The Central Dogma of Molecular Biology

Source: Access Excellence

Figure 5: The Central Dogma of Molecular Biology

The ultimate goal in understanding the human genome is to identify how key regulatory networks lead to the synthesis of proteins that sustain life processes. Understanding how these processes contribute to life provides a foundation for understanding how things can go wrong and cause disease. Additionally, by investigating how certain networks behave under stress or stimulus, a scientist can learn much about how the human body combats disease. In most cases, disease states are characterized by a malfunction in the genome. In cancer, for example, certain networks have gone awry and cause the cells to multiply uncontrollably. Understanding the building blocks and pathways of each network begins with a more complete grasp of how and when life's blueprint (DNA) is regulated.

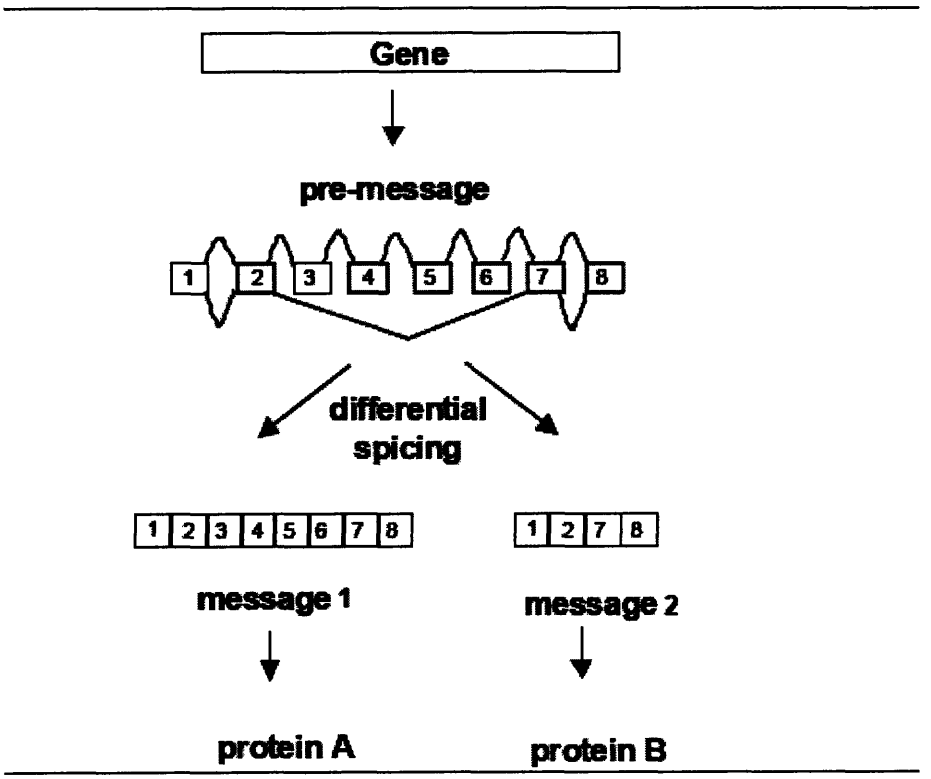
3.1.1 DNA Sequence Data

The advent of DNA sequencing tools has allowed scientists to precisely determine the entire genome complement of many organisms. However, knowledge of the code represents just the start of understanding biological processes. Of the three billion base pairs that comprise the human genome, only 3%-5% of the sequence represents genes, with the remainder considered "junk DNA." Additionally, each gene is divided into multiple regions: (1) exons (responsible for coding proteins) and (2) introns (or non-coding regions). By deploying bioinformatics tools, scientists can more readily separate genes from junk DNA and exons from introns. Although machines have automated much of the process, it remains laborious, and just 33% of the human genes out there have been located.

3.1.2 Gene Identification

To properly identify a gene's sequence scientists must (1) recognize the start site, (2) identify non-coding regions, and (3) characterize the end point. All three of these descriptive components are nearly impossible without the appropriate data analysis tools to sift through reams of four-letter data (e.g., ATCGCGATCGCGATAT). Additionally, regions of DNA that exist before and after genes can play a role in the execution of regulation. These elements must be identified and annotated, as well. In other words, computer programs, like the cell,

must find the beginning and end of the gene, recognize the exons, cut out the introns, and paste together all the exons that encode for a given protein. Adding complexity to the process is the factor of differential splicing, where certain intron-exon boundaries within the message are skipped altogether. In doing so, the same gene has the potential to create a multitude of proteins that carry out different functions. For example, differential splicing events can produce three separate protein products, which are all encoded by one gene whose message is mixed and matched, as the cell requires.



Source: UBS Warburg LLC

Figure 6: Gene Splicing

3.1.3 Gene Identification and Annotation Tools

There are a variety of gene sequence analysis tools that are available to public and private research communities. For example, gene identification programs like GRAIL and Genefinder are freely available online and provide researchers with minimal elements of gene analysis. Using these programs, a researcher can submit a

DNA sequence of interest, and in a short period of time, a description of the gene is returned (assuming it has been previously "annotated" by another researcher). Although it is difficult to criticize tools that are free, we should note that these online applications often lack the robust functionality and security of commercial software. Furthermore, because annotated genes are stored in disparate locations, researchers are often forced to submit their findings to a multitude of Web sites (not the most efficient process).

In summary, the sequencing efforts of today will be fueled by a combination of both public and private efforts. Unfortunately, the biological data generated from this effort remains not only vast, but is stored in a multitude of locations. Clearly, sifting through this mountain of data requires information technology.

Going forward, aggressive sequencing efforts will create a "snowball effect," with larger data sets, more exact annotations, and a better understanding of regulatory networks. Most important, this information should provide the scientific community with more targets to screen advanced chemical compounds. Today, there are roughly 10,000 human genes for which a function has been determined. Of these, approximately 500 are currently targets for synthetic drugs designed to combat the diseases that are associated with them. The number of available targets could increase fivefold with the deployment of bioinformatics platforms designed to accelerate comparative genomics, functional genomics, structural genomics, pharmacogenomics, and proteomics.

Database Sequence Searching

The following steps outline the typical process of performing a database sequence search.

(1) *Choose a novel or known sequence to be analyzed.* The researcher may be looking for information regarding gene family relatives, functional annotation associated with the sequence, or validation of a hypothesis.

(2) *Determine which server to use for the query.* The researcher may conduct the sequence similarity search against many public or proprietary servers.

(3) *Select a program available on the server that is relevant to the search.* Some programs look specifically for text, others search through all available data. In addition, most programs are specific to the type of sequence being queried (i.e., amino or nucleic acid).

(4) *Select a filter to limit the number of false positive hits.* Statistical parameters can be set to increase or decrease the relative similarity between the query and the database to return matches with varying degrees.

(5) *Analyze the output file*

3.1.4 Public Search Tools

When data is gathered from a number of sources, data mining tools can identify important information, trends, or examples from the data set. These publicly available tools might include statistical models, rules-based procedures, interactive visual manipulations, or combinations of these or other techniques. Users must also be able to look at and manipulate data sets, identify trends and outliers, and test statistical models or rules. A number of data search tools have been developed for public use to help researchers sift through the enormous amounts of genomic information. Some of these include BLAST, Entrez, and FASTA.

BLAST. BLAST (basic local alignment search tool) was introduced in 1990 as part of a suite of DNA- and protein-sequence search tools. This software, available through The National Center for Biotechnology (NCBI), allows researchers to customize their searches to compare an amino acid or nucleotide sequence to databases of sequences. BLAST uses an approximation method that is expected to find matches quickly and with a statistical measure of significance (that infers a biological connection). However, this rapid process sometimes leads to misleading matches. The BLAST server is most easily accessed via the Internet, through NCBI's Web site.

Entrez. Developed by NCBI in 1991, Entrez provides access to sequences of nucleotides that are linked to proteins. This search tool, now accessible through the NCBI home page on the Internet, enables researchers to sift through information stored in its databases, including text, 3-D molecular structure,

genome maps, and phylogenetic taxonomy. Once a search is submitted, a summary of each hit is returned, which can be viewed in several different formats.

FASTA. FASTA, maintained by the EBI in the United Kingdom, was the first widely used algorithm designed for similarity searches within databases. This engine scans sequences for tiny matches that will identify an optimal local alignment (an alignment of a portion of two nucleic acid or protein sequences with the highest possible score).

3.2 Comparative Genomics

Comparative genomics is the practice of comparing a gene or protein sequence with the sequence of another gene or protein. Depending on the degree of similarity that exists, scientists can use this information to extrapolate functional and evolutionary relationships. In theory, the more two sequences resemble each other, the greater the probability that they perform similar functions and are evolved from a similar ancestor.

In practice, predicting the function of "unknown" genes requires a similarity search against genes with "known" functions. To do this, a newly sequenced region of DNA is used as bait to "fish out" genes that resemble its primary sequence. If this screen is successful in "hooking" previously annotated genes (with known functions), then the scientist is provided a clear direction for future experiments. Although these predictive methods have been around for some time, the exponential growth in genomic databases (and known genes) will fuel explosive growth in this science. After all, with nearly 10,000 human genes and numerous genes from other organisms discovered to date, the likelihood of finding a sequence similarity continues to increase. Moreover, by understanding the functions of related genes, scientists can narrow the focus of their research more quickly and efficiently.

Although the industry has made substantial progress in recent years, there remains a gap between sequence data and functional understanding. (See Figure 7.) More aggressive efforts to screen and compare known and unknown genes should facilitate a more complete understanding of sequence inter-relationships.

Using the science of comparative genomics, researchers can often locate the mouse counterpart to the human gene known to be involved in a disease. Moreover, knock-out and knock-in mouse technology (the process of altering the genome of the mouse prior to birth to physically remove any gene or add an extra copy of a gene) provides the ability to simulate human disease in a mouse. Once altered, these model organisms can be studied and analyzed for their responses to various drug treatments. Cancer is one example of a disease that has been replicated in a model organism for the purposes of studying the biology and testing potential therapeutics. Beyond testing efficacy, model organisms are also beneficial in understanding toxicity. As mentioned previously, drug discovery ventures prefer to understand "negative reactions" earlier rather than later, as this can save many lives and millions of dollars. Producing animal models of human diseases allows drug candidates to be tested in a living organism. This approach will allow drug companies to "fail early and often" and focus research and development budgets on only the best drug candidates.

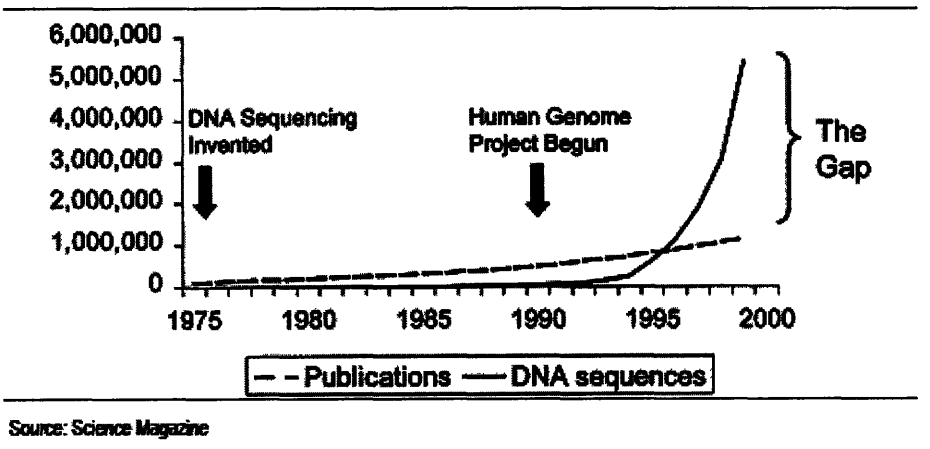
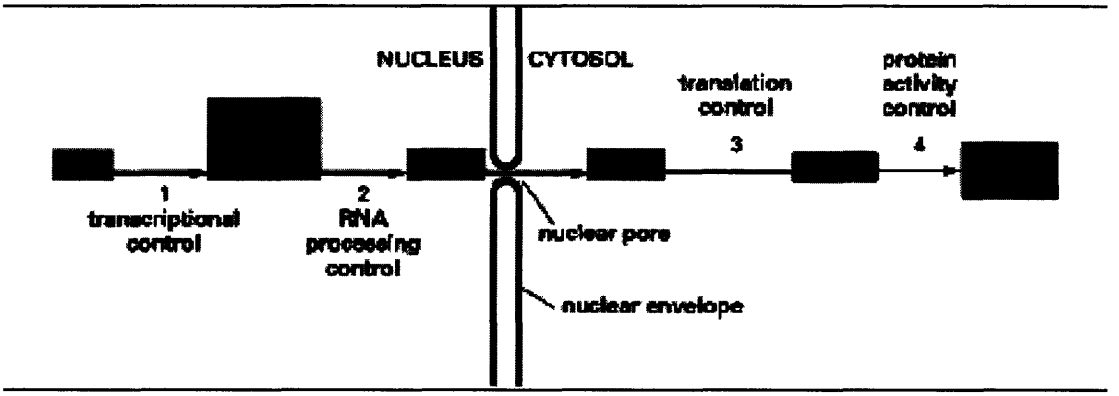


Figure 7: Raw Sequence Data Versus Functional Annotation

As mentioned previously, informatics plays a key role in accelerating comparative analyses and assessing end results. Examples of third-party software used for comparative genomics analysis are LION Bioscience's GenomeSCOUT (Figure 8) and GeneData's GD Phylosopher.

Beyond third-party applications, researchers also have the option of performing comparative analyses online. For example, NCBI provides Internet links to basic



Source: Access Excellence

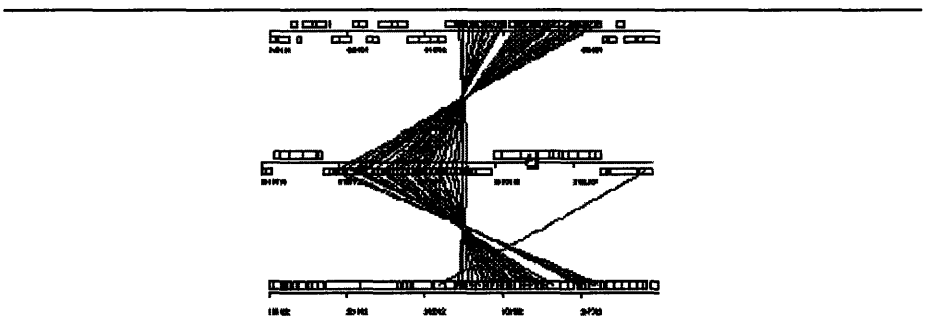
Figure 9: The Multiple Control Points of Gene Expression

Although our initial discussion emphasized the importance of sequence data in providing a "road map" to the human genome, many believe that the process of transcription holds the key to understanding "how and why" genes are activated. More important, the expression of one gene often catalyzes positive and negative transcription activity in multiple genes. By following this chain reaction, researchers can begin to comprehend the intricate complexities of regulatory networks and diseases in humans and animals. This approach could also provide scientists with multiple targets to inhibit diseases with drug compounds. For example, if Type II diabetes is caused by a chain reaction of genes A-Z, then researchers have 26 possible targets for halting the "domino effect" and treating this condition.

To analyze the transcription process in greater detail, the genomics community has developed a variety of methods to assess which genes are expressed or "turned on" under controlled and specified conditions. Today, the most common technique for measuring RNA quantity is through the use of microarray technology. A DNA microarray consists of large numbers of DNA molecules spotted in a systematic order on a solid support (glass slide, nylon membrane, or silicon chip). The DNA sequence at each spot represents a single-stranded fragment of a gene in the genome. The short length (around 25-75 bases) and the single-stranded nature of these DNA molecules allow them to pair with their complementary partners if they were to come in contact with them.

comparative analysis software. Most of these programs, like BLASTX and ClustalW, require the researcher to electronically submit their raw sequence data to an unsecured outside server where the analysis takes place. As with most online tools, data outputs are crude and typically not formatted for presentation purposes or integration with other data sets. Additionally, these resources only search GenBank, the largest public genomic database.

Although online resources are sufficient for academic institutions, most for-profit organizations have embraced more robust on-site tools. This not only enhances data search capabilities but also provides for further integration with enterprise-wide bioinformatics platforms.



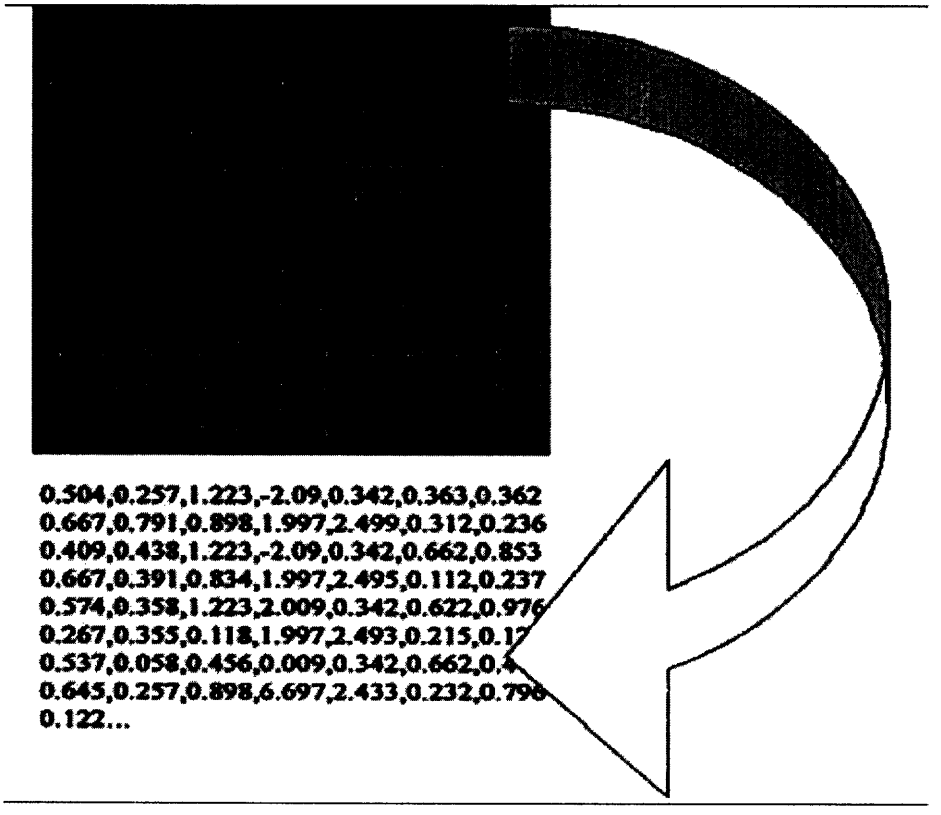
Source: LION Bioscience

Figure 8: LION Bioscience's GenomeSCOUT

3.3 Functional Genomics

For a human gene to perform its prescribed function, its DNA sequence must be expressed (copied) into an RNA message through a process called "transcription." Once this process takes place, the RNA message moves outside the cell nucleus and is translated into a protein. From a functional genomics perspective, DNA represents the "instructions for life"; RNA acts as the "messenger"; and proteins carry out various body "functions" necessary to sustain life (e.g., metabolism, growth, and fighting off diseases or infections).

Utilizing the aforementioned technology, scientists can quickly assess differential expression levels for multiple samples. For instance, by finding which genes are improperly turned on in a diseased tissue, researchers can quickly narrow their list of potential targets for therapeutics. Although new microarray technology has addressed the challenge of data collection, it has also created a new bottleneck - data analysis. In many cases, researchers still upload experimental data into Excel-based spreadsheets and sort through results using low-end tools. With this archaic approach, the knowledge extracted from gene expression data is clearly dependent on the scientist's ability to formulate "home-grown" algorithms for analysis. Once all data points are collected, as shown in Figure 10, this information is translated into a "gene expression matrix," with each column representing a gene and each row representing a unique sample (e.g., different treatments or different growth stages). Expression analysis software allows the researcher to evaluate entire microarrays, or narrow their search to just a subset of genes.



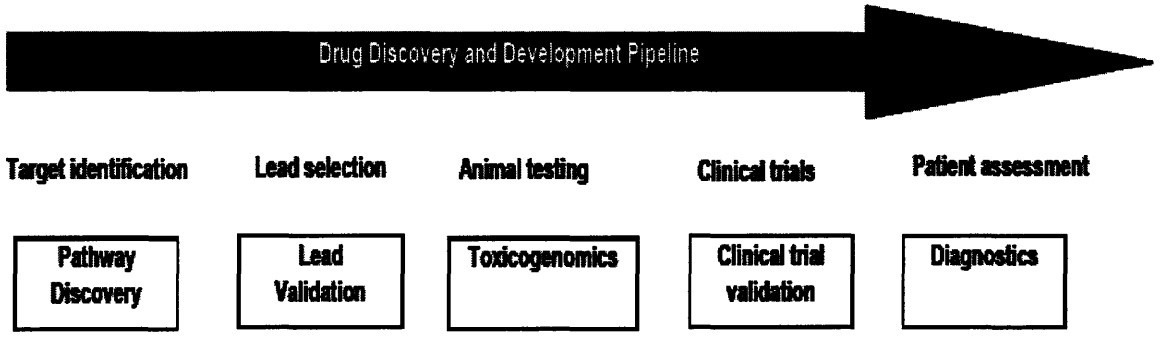
Source: Spotfire

Figure 10: Digitizing Microarray Images

Examples of higher-end, multi-functional gene-expression packages include Rosetta Resolver, Silicon Genetic's GeneSpring and SpotFire's Decision Site®.

3.3.1 Expression Analysis In Discovery

The power of gene expression analysis can be applied to the entire drug discovery process, from the initial step of target identification to the latter phases of clinical trial optimization. Today, most gene expression experiments are designed to identify which genes get turned "on" and "off" under a given condition, which is beneficial in the identification of potential drug targets. Although this represents an effective application of this science, it has broader utility across the entire spectrum of drug discovery - from metabolic pathway determination to diagnostics (See Figure 11).



Source: UBS Warburg LLC

Figure 11: Applications for Gene Expression in Drug Discovery

The following describes these applications in further detail: Determination of cellular regulatory pathways. Since proteins originate from an RNA message, researchers believe that a complete and thorough understanding of cellular circuitry (i.e., when and why each message is made) can be developed through gene expression profiling experiments. By aggregating this data, we believe the scientific community can gain a better understanding of how a cell's regulatory pathways function to sustain life.

Target identification. One of the main barriers in developing therapeutics has been the identification of rational targets against which to design drugs. This barrier is more easily traversed with a gene expression profiling approach that integrates the proper bioinformatics tools. For example, once we understand how regulatory networks function under normal circumstances, scientists can begin to infer how regulatory malfunctions can lead to disease. Employing gene expression experiments to uncover differences in healthy and diseased tissues can clearly identify cellular circuitry that has gone awry. By studying these situations, scientists can begin to narrow the list of potential drug targets for a given disease. Target validation. Most drugs are designed to combat specific protein targets whose presence is somehow involved in causing disease. With target identification, the researcher can reveal which aspects of the regulatory pathways are malfunctioning. With target validation, the researcher adds a candidate drug to a diseased cell and assesses its effect with gene expression profiling. For example, by administering the potential drug, the researcher can determine whether the malfunctioning networks have been repaired and if any side effects occurred. By validating efficacy, researchers have a quantitative means of choosing which potential compounds proceed to the next stage of drug discovery.

Toxicogenomics. This is an emerging discipline that identifies the adverse effects of chemicals or physical agents on biological systems. Using gene expression profiling to measure response rates on a molecular level, researchers can identify known and suspected toxicants (adverse effects). Physiologically, the way a body responds to potential hazardous chemicals (e.g., turning on gene A and turning off gene B) is of great interest to drug discovery ventures. For example, many cancers have been linked to exposure to harsh chemicals or carcinogens. By understanding toxicological effects before drugs move on to the next step, pharmaceutical companies should be able to dramatically improve success rates of clinical trials and lower drug development costs.

Clinical trials validation. Clinical trials provide a "real life test" of potential drug candidates. After a period of time, researchers can assess the impact of the tested products in terms of efficacy and adverse side effects. To accelerate this process,

gene expression analysis could be conducted on clinical trial patients to evaluate efficacy at the molecular level, before outcome studies are produced. In addition, dosage and administration of the candidate drug could be optimized, and patients with differential response rates could be classified by their genetic makeup (i.e., single nucleotide polymorphisms, or SNPs).

Diagnostics. Diseases are often caused by malfunctions in molecular pathways. For example, cancerous tissues have lost their ability to turn off the "pathway" leading to cellular division and therefore multiply uncontrollably. Today, regulatory malfunctions can often be detected with gene expression profiling. As more regulatory networks are identified, physicians will be able to compare a patient's gene expression profile with a database of typical disease profiles and make a comparative inference. This methodology has already been proved effective in diagnosing two forms of leukemia. This facet of gene expression profiling will be instrumental in providing doctors with a more accurate medical diagnosis and allow for the administration of the correct therapeutics.

3.4 Pharmacogenomics

Pharmacogenomics is the study of how distinct groups of individuals respond to therapeutics as a result of differences in their genome. It was recently discovered that small inherited differences in DNA sequences of individuals may confer an increased susceptibility to disease and may even dictate whether some chemicals will work to combat disease while others may be toxic. Although more than 99% of human DNA is the same across the world population, slight sequence variations can have a major impact on how each human responds to bacteria, viruses, toxins, chemicals, drugs, and other therapies. These differences are termed single nucleotide polymorphisms, or SNPs. Since drugs mainly act to block the function of some protein involved in disease, small variations in the genome can influence whether the drug will act positively, negatively, or not at all. Differential responses to chemotherapy, for example, can now be pinpointed to specific "markers" of the patient. A marker is simply defined as a minor change in the genome (SNP) that correlates to a physiological response. These DNA

variances are translated into small structural differences in the protein, which may influence its ability to interact with small molecules. By identifying these markers in all diseases (and in all populations), the life sciences community has the opportunity to tailor medications to the individual to ensure that the most effective, targeted treatment is administered.

To identify SNPs associated with specific diseases, a handful of companies are tackling the challenge of population genomics. Unlike most bottom-up approaches to life sciences, population genomics starts with a group of patients diagnosed with a specific disease, and thereafter, tries to find the unique characteristics that the disease group shares in common. According to theory, SNP variations are responsible for many common diseases, providing researchers with a handful of targets to inhibit protein function.

One of the early leaders in this field is deCODE Genetics, which is cooperating with the government of Iceland to screen medical records and extract genomic information from the country's entire population. Given the isolated nature of this country and the inheritability of SNPs, deCODE will gain valuable insight from studying this controlled environment.

Today, SNP research is largely focused on two areas - discovery and scoring. In terms of discovery, the SNP Consortium, composed of both academic and commercial partners, recently announced a goal of identifying 300,000 markers in the coming year. Once these markers are properly classified, researchers will begin the process of identifying or "scoring" patients with specific SNPs. This "scoring" system promises to assess the likelihood that an individual will develop a given disease or respond to a specific drug.

The technology for scoring SNPs is also emerging quite rapidly. High-throughput techniques that can identify small variances in DNA can warn doctors and patients of increased susceptibilities to fatal diseases. In addition, data collection is made easy with today's microchip, mass spectrometry, and electrophoresis-based tests, with laboratories requiring just a single drop of blood to perform a full diagnostic screen.

Some players in this field are Silicon Genetics (the company's Allele Sorter is capable of sorting and analysing large quantities of SNP data), Incyte Genomics (the company's Snooper mined proprietary database for single nucleotide changes) and decode Genetics (the company has developed an advanced solution for population genomics).

3.5 Structural Genomics

Structural genomics is the study of a protein's three-dimensional shape and how this ultimately correlates to function. A malfunction or disruption of one or a combination of a protein's functions in a regulatory network can lead to disease. Knowing how these networks proceed in healthy tissues provides a foundation for understanding how disruptions in these networks can lead to disease. A full understanding of a protein's three-dimensional structure and how it interacts with other molecules can give researchers a "visual target" for designing drugs to inhibit protein function (and stop the spread of disease).

The potential for therapeutics, derived from structural data, is immense. For example, a thorough structural analysis of targeted proteins can minimize unanticipated (adverse) side effects that "pop up" in clinical trials. Structural assessments can also facilitate rational drug design, specifically tailored to the patient. As such, this emerging science will serve as a critical component of any successful drug discovery venture. Moreover, as structural genomics gains momentum in the "mind share" of researchers, we anticipate a corresponding boost in demand for visualization software (a requirement for tracking and manipulating three-dimensional protein structures). To determine the structure of a protein, it must be replicated in large quantities to ensure an adequate sample size. To do this, specifically engineered bacteria are injected with a DNA sequence that corresponds with a specific protein. These bacteria are then stimulated to produce large quantities of the corresponding protein. Thereafter, proteins are purified through biochemical methods and structurally analyzed by one of three methods (X-ray crystallography, cryo-electron microscopy, and NMR). Alternatively, efforts to bypass these wet lab approaches

and convert DNA sequence data into a three-dimensional protein molecule at the desktop have also begun in a field called theoretical modeling. We shall discuss each of these four disciplines in greater detail in the paragraphs that follow. Although this science focuses on the structural aspects of proteomics, it remains highly dependent on the continued growth and annotation of genomic data.

On the software side of the equation, a number of informatics vendors have developed programs to: (1) expedite the mathematical calculations needed to translate images (or sequences) into structure; and (2) visualize protein structures as a means of determining functions. Some of the prominent vendors in this sector are: Molecular Structure Corporation (the company's d*TREK software automates the process of finding X-ray diffraction spots and predicting relative distances between the nuclei of atoms), Molecular Simulation (a subsidiary of Pharmacopeia) and Structural Bioinformatics (the company's proprietary software employs theoretical modeling techniques to generate three-dimensional structures from sequence data).

3.5.1 Comparative Analysis

It is common for researchers to perform comparative analyses between two proteins. The reason is that most proteins belong to distinct families whose structures are highly similar, with slight variations separating one protein function from another. Interestingly, some proteins have different sequences but still fold into remarkably similar structures. Although most protein families can initially be identified at the protein sequence level and are confirmed at the structural level, proteins that fold into similar structures but have no sequence conservation can only be identified at the structural level. Understanding how different structural modules contribute to function allows researchers to extrapolate an unknown protein's function by comparing it with previously identified proteins. Moreover, using statistical parameters and mathematical probabilities, algorithms can search through structural databases for similarities not visible to the naked eye. Obviously, bioinformatics represents an integral part of these complex structural analyses.

It should be noted that many drugs on the market have varying effects on different patients. For example, one patient might respond positively to Claritin, while another may continue to suffer from allergies. The basis for this variability is a drug's inability to interact with its protein target due to a slight structural change that arises from an SNP in the patient. Through structural genomics, scientists can now understand how these slight variations in the DNA code translate into structural protein changes that influence how and why a drug does not interact with its protein target. Assessing protein structure with sophisticated bioinformatics platforms could allow researchers to design more efficacious, patient-specific drugs while limiting adverse side effects.

3.6 Other Life Science Informatics Categories

Among the other life science informatics categories are Image Informatics, Cheminformatics, ADME/Tox and Clinical Trial Informatics. Image informatics deals with reading images generated from cells and tissues and acquiring data from them in a digital format, storing it for future reference, and seamlessly accessing it to correlate past and present experiments. Cheminformatics is based on an optimization technique to combine computational chemistry with three-dimensional structure analysis, enabling fast and easy synthesis of compounds *in silico* according to the researcher's needs. Examples of some leading companies that provide cheminformatics solutions are Tripos, MolSoft. Beyond drug discovery, informatics can be used in pharmacokinetics to model the release and body-uptake of drug molecules in the body and in clinical trials to design experiments on patient population and analyze results.

4 Market Landscape

As detailed in the previous section, there are boundless opportunities for the application of information technology in the drug discovery process, and that it represents the next quantum leap in drug discovery. According to UBS Warburg Investors report, the Bioinformatic market was approximately $700M in 2001 and is anticipated to reach $1.7B by 2006 growing at 20% annually.

4.1 Market Segments by Product Area

The market for bioinformatics can be segmented into three categories on the basis of product area – Analysis software, Content and database providers and the IT Infrastructure segment. The analysis segment is expected to grow the fastest at 26% followed by the IT Infrastructure segment (19%) and then the Content segment (15%) (Table 2). Table 3 shows a list of the market players in these segments and their key financial statistics.

Players in the Analysis segment provide data analysis and statistical packages for applications such as gene expression analysis, sequence analysis, image data processing, sequencing software, etc. The product can be either stand-alone (as in the case of DNASTAR, Inc.) or a web-based application-service (as in the case of BioDiscovery, Inc.). Other examples include Compugen, Celera Genomics, Rossetta, etc.

Content source providers host biological databases (such as Genbank, SNP Consortium, Protein Data Bank, etc.). They can be public, in-house or commercial. While companies will continue to allocate significant dollars toward purchasing commercial content, increasing availability of public data will affect reliance upon commercial data in the future. The large volume and relatively comparable quality (arguably so) of data in the public domain will rival the cost of purchasing data from the commercial sector. Internal data generation and integration will result in increased reliance upon in-house content. Examples of content providers are Gene Logic, Compugen and Celera Genomics.

A third category of lifesciences software providers are the enterprise software companies that provide system-wide management tools for laboratories, research units, etc. These tools enable research organizations to store, manage, integrate, and analyze large amounts of genomic and proteomic data from disparate sources. By providing a common interface for all data, multiple users can share results, and research and development managers can coordinate projects effectively. With these solutions, organizations can also leverage prior investments in informatics applications while allowing for the purchase and integration of advanced third-party tools. Examples of players in this field are Tripos, Pharmacopeia.

Year	Market Size	Mkt Segmentation by Product Area			Mkt Segmentation by Application Area			
		Analysis Software	Content Source	IT Infrastruc-ture	Genomics	Proteo-mics	Pharmaco-genomics	Cheminfo rmatics
2001	697	202	225	270	383	90	70	153
2002	836	254	258	325	383	126	97	231
2003	1004	319	296	389	383	175	134	312
2004	1204	401	339	464	383	243	184	394
2005	1445	504	388	553	383	337	254	470
2006	1734	634	445	655	383	469	351	531

Table 2: The Market for Bioinformatics in Drug Discovery and Development

4.2 Market Segments by Area of Application

The bioinformatics market can also be categorized on the basis of application areas, namely genomics, proteomics, pharmacogenomics and cheminformatics (each of these categories were explained in detail in Section 2). While genomics has the majority market share, proteomics and pharmacogenomics markets are expected to grow the fastest at 39% and 38% respectively (Table 2). Many bioinformatics companies operate across all four or at least 3 of the above markets. Some exclusive players in cheminformatics and pharmacogenomics also exist.

	Analysis Software		Content	Enterprise
	ASP-Based	Stand-Alone		
Accelrys		X	X	
BioDiscovery	X	X		
Celera		X	X	
Compugen	X		X	
Curagen	X		X	
DNASTAR		X		
DoubleTwist	X		X	
Gene Codes Corporation		X		
Gene Logic		X	X	X
GeneData			X	
Partek		X		
MDL				
Incyte			X	
Rosetta				
Lion Biosciences		X		X

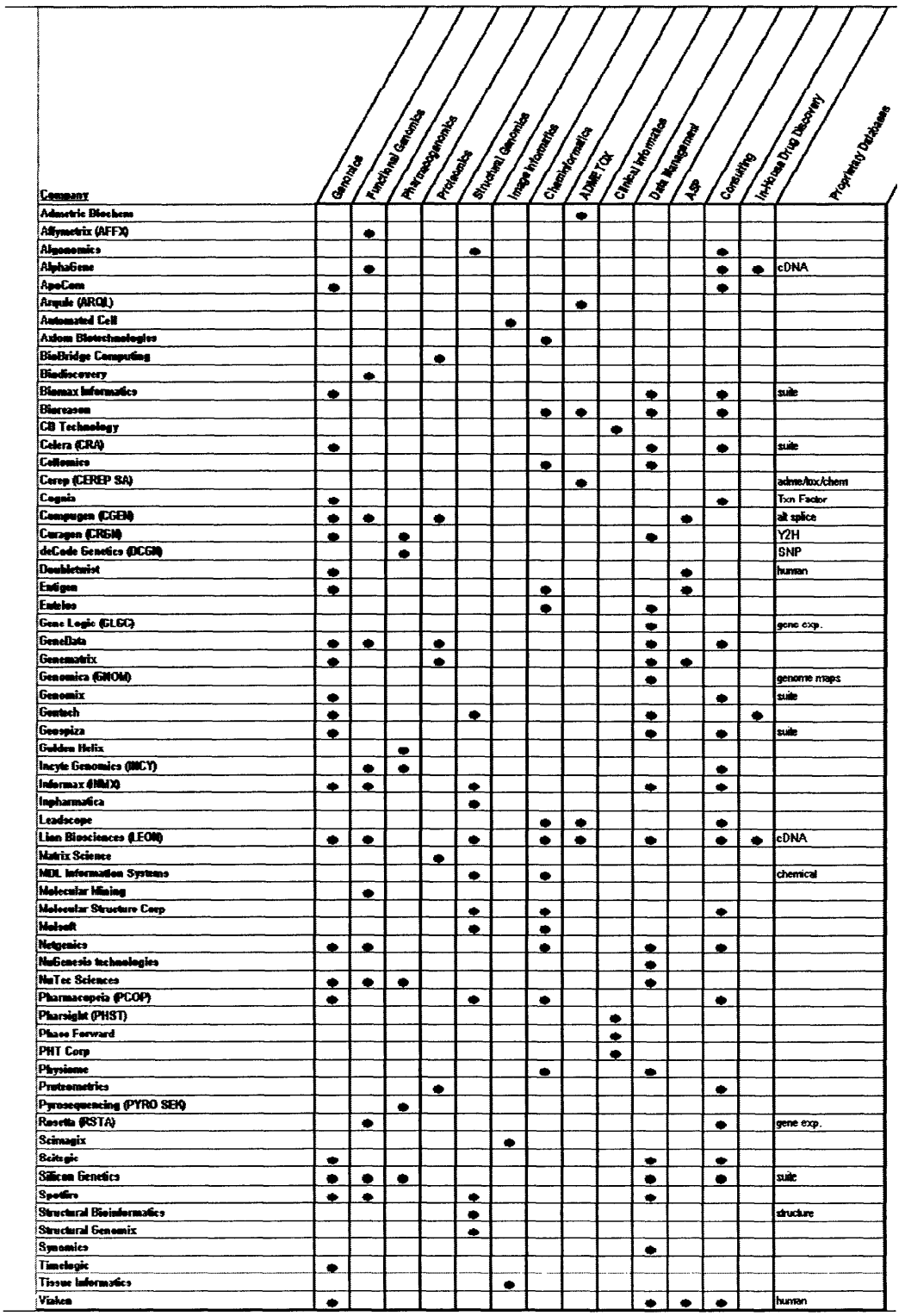
Table 3: Leading Suppliers by Product Area

4.3 Competitive Landscape

Given the recent emergence of third-party informatics competitors, it is somewhat challenging to characterize the competitive landscape. Low capital requirements (and barriers to entry) have created a highly fragmented environment. To add to the confusion, pure-play vendors face competition from a variety of angles, including nonprofit organizations, in-house solutions, and traditional technology companies. The following discussion will attempt to provide a framework for the current environment.

Today, the bioinformatics universe consists of six major public participants (Compugen, CuraGen, Genomica, InforMax, LION Bioscience, and Rosetta) and more than 100 privately funded start-ups. Because bioinformatics technologies are not one size fits all, each competitor attacks the market in a slightly different manner, with competition generally divided up by scientific discipline (e.g., pharmacogenomics, proteomics, and so forth). For example, Rosetta Inpharmatics specifically addresses computational needs of functional genomics (gene expression profiling), while ProteoMetrics offers solutions for the field of proteomics (mass spectrometry data acquisition and mining). Figure 12 shows the competitive landscape in greater detail.

Although most companies have developed best-of-breed applications for specific disciplines, a handful of vendors have created enterprise-wide solutions to manage projects (and data) throughout the entire life cycle of drug development. An example of this strategy is LION Bioscience, which provides an integrated, "one-stop-shop" offering for data management, genomics, functional genomics, proteomics, and chemistry. Most pharmaceutical companies are evolving toward this multi-disciplinary approach, a trend that should favor the early leaders in this space (e.g., LION Bioscience and InforMax).



Company	Genomics	Functional Genomics	Pharmacogenomics	Proteomics	Structural Genomics	Image Informatics	Cheminformatics	ADME/TOX	Clinical Informatics	Data Management	ASP	Consulting	In-House Drug Discovery	Proprietary Databases
Adnetric Biochem								●						
Affymetrix (AFFX)		●												
Algonomics					●							●		
AlphaGene		●										●	●	cDNA
ApoCom	●											●		
Arqule (ARQL)								●						
Automated Cell						●								
Axiom Biotechnologies							●							
BioBridge Computing				●										
Biodiscovery		●												
Biomax Informatics	●									●		●		suite
Bioreason							●	●		●		●		
CB Technology									●					
Celera (CRA)	●									●		●		suite
Cellomics							●			●				
Cerep (CEREP SA)								●						adme/tox/chem
Cognia	●											●		Txn Factor
Compugen (CGEN)	●	●		●							●			alt splice
Curagen (CRGN)	●		●							●				Y2H
deCode Genetics (DCGN)			●											SNP
Doubletwist	●										●			human
Entigen	●						●				●			
Exelexis							●			●				
Gene Logic (GLGC)										●				gene exp.
GeneData	●	●		●						●		●		
Genematrix	●			●						●	●			
Genomica (GNOM)										●				genome maps
Genomix	●											●		suite
Gentech	●				●					●			●	
Geospiza	●									●		●		suite
Golden Helix			●											
Incyte Genomics (INCY)		●	●									●		
Informax (IMAX)	●	●			●					●		●		
Inpharmatica					●									
Leadscope							●	●				●		
Lion Biosciences (LEON)	●	●			●		●	●		●		●	●	cDNA
Matrix Science				●										
MDL Information Systems					●		●							chemical
Molecular Mining		●												
Molecular Structure Corp					●		●					●		
MolsoR					●		●							
Netgenics	●	●					●			●		●		
NuGenesis technologies										●				
NuTec Sciences	●	●	●							●				
Pharmacopeia (PCOP)	●				●		●					●		
Pharsight (PHST)									●					
Phase Forward									●					
PHT Corp									●					
Physiome							●			●				
Protometrics				●								●		
Pyrosequencing (PYRO SEQ)			●											
Rosetta (RSTA)		●										●		gene exp.
Scimagix						●								
Scitegic	●									●		●		
Silicon Genetics	●	●	●							●		●		suite
Spotfire	●	●		●						●				
Structural Bioinformatics				●										structure
Structural Genomix				●										
Synomics										●				
Timelogic	●													
Tissue Informatics						●								
Viaken	●									●	●	●		human

Source: UBS Warburg LLC

Figure 12: The Informatics Landscape

4.3.1 Indirect Competitors

As mentioned previously, pure play bioinformatics vendors face indirect competition from a variety of angles, including nonprofit organizations, in-house solutions, and traditional technology companies. The following describes these market participants in greater detail:

Nonprofit organizations. Over the past decade, a number of publicly available databases have developed "low-brow" bioinformatics solutions. Examples would include the sequence search engines, such as BLAST (available through GenBank) and FASTA (available through SwissProt). Although we expect these ASP-based products to compete on the lower end of the market, they do not represent a serious competitive threat to the more robust bioinformatics solutions.

Home-grown bioinformatics solutions. Since most third-party bioinformatics vendors arrived late to the game, many large pharmaceuticals were forced to build in-house bioinformatics departments to manage discovery projects and analyze data. Companies in this category would include GlaxoSmithKline and Wyeth-Ayerst. Although these home-grown solutions represent a near-term competitive threat, pharmaceutical companies will ultimately realize the benefits of outsourcing these initiatives to third-party vendors.

Large IT companies. A third competitive pressure exists from large information technology companies that have recognized the demand for powerful applications in life sciences. Since many of these organizations already maintain established relationships with the pharmaceutical industry, capitalizing on the genomics revolution with new products and services is a natural fit. The following details the strategies of three recent entrants to the life sciences market.

IBM. IBM has a suite of offerings for the life sciences community. Discoverylink, its database integration tool, is a middleware application that links external life science databases with disparate formats. The company also offers Internet hosting capabilities and IT consulting services. To date, the company has formed strategic alliances in data integration (NetGenics),

genomics (Incyte), proteomics (MDS Proteomics), and structural genomics (Structural Bioinformatics, Inc).

Compaq. Compaq assists drug discovery ventures (via supercomputers) in processing statistical calculations relating to gene similarities and gene expression profiles. To date, the company has formed a handful of strategic partnerships with genomic database providers and informatics solution vendors. For example, the company's hardware was instrumental in providing Celera with the necessary computing power to complete the human genome project.

Silicon Graphics. SGI provides life science computational support through a variety of scalable solutions (from the personal workstation up to the supercomputer). This equipment accelerates the speed at which genomic data is converted to useful knowledge. For example, Protherics is taking advantage of SGI's powerful computing platforms through a DOCKCRUNCH program that virtually screened one million compounds in six days. As the industry evolves toward larger, more complex data sets, we believe SGI will play an instrumental role in data processing support.

4.4 Market Trends

4.4.1 Consolidation on the horizon

Looking ahead, as information technology plays a more critical role in drug discovery, we expect significant consolidation in this cottage industry. In our opinion, multi-disciplinary drug discovery programs (which encompass all scientific disciplines) will become the rule and not the exception. In this environment, organizations will require full compatibility between tool manufacturers, software vendors, and proprietary in-house solutions. The following suggests some possible events that could spark another wave of M&A activity.

Intra-industry consolidation. Pure play software vendors may look to integrate and offer combined solutions through acquisitions and alliances. Pharmacopeia's acquisition of Oxford Molecular Group is one example where a cheminformatics solution provider acquired a genomics analysis module to

increase the breadth of their solution. Similar partnerships, designed to merge biology with chemistry, are on the horizon.

Forward integration by tool and content companies. Tool companies and content providers may seek to enhance the value of their current products by integrating innovative, value-added technology. For example, Affymetrix's acquisition of Neomorphic allowed the leading distributor of microarrays to integrate computational genomics and sophisticated bioinformatics into their product pipeline.

Screening companies enter the foray. High-throughput screening companies that perform contracted services to pharmaceutical companies may look to increase the breadth of their offering by adding an informatics component. One example is Discovery Partners' acquisition of Structural Proteomics, which added a powerful in silico protein analysis program to its existing chemical screening capabilities.

Large information technology companies stepping up to the plate. A fourth - and less likely - scenario would be the acquisition of pure-play informatics vendors by large information technology companies. This would allow these conglomerates to further penetrate this vertical market (with potentially explosive growth opportunities).

4.4.2 In-house vs. third-party software development

One of the biggest questions facing users of bioinformatics is whether to purchase content or technologies from commercial vendors, or to develop tools in-house to meet the company's specific needs. Front Line Strategic Consulting estimates that pharmaceutical and biotechnology companies will continue to allocate 60% of their total bioinformatics spending to commercial vendors, totaling $1.1 billion in 2006. According to UBS Warburg LLC, majority of technology investments in 2001 were performed in-house rather than outsourced to third-party vendors (e.g., LION Bioscience and InforMax). Going forward, however, the pharmaceutical

industry is expected to realize the benefits of informatics partnerships and fundamentally shift resources toward more robust third-party technology.

5 Business Models

Bioninformatics vendors generally employ one of three business models as detailed below. Although the opportunity for informatics appears relatively straightforward, we should emphasize that each competitor attacks the market in a slightly different manner, with competition and products typically bifurcated by specific scientific disciplines (e.g., genomics, functional genomics, proteomics, and so forth). In addition, the delivery of these products and services varies from vendor to vendor, with each employing a slightly different delivery mechanism and business model. The following sections describe the three most popular business models in this nascent market as well as the advantages (and disadvantages) of each approach.

5.1 Application Service Providers

Initially, a number of vendors entered the informatics market as application service providers, or ASPs. At first, many thought this data delivery approach was ideal for pharmaceutical companies looking to minimize capital allocations, cut overhead costs, and lower the risk of technological obsolescence. In addition, the ASP model provided predictable technology costs for clients (with bundled services billed monthly), while enhancing recurring revenues (top-line predictability) for informatics investors. This emerging class of .portal. companies included a handful of well-known start-ups such as DoubleTwist, Viaken, and Entigen.

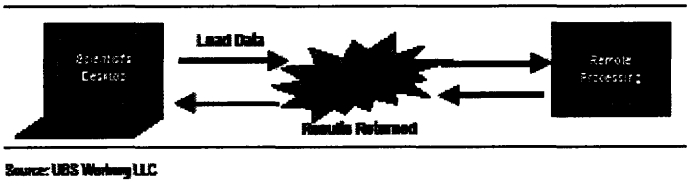


Figure 13: The ASP Solution

Although this approach made sense in theory, many start-up companies failed to realize the privacy concerns of large pharmaceutical companies. As mentioned previously, because the ASP model depends on the Internet for data delivery, large biopharmaceutical organizations were naturally concerned about security. After all, with the average multi-national pharmaceutical company spending hundreds of millions of dollars on drug discovery, it simply doesn't make sense to float these trade secrets into the public domain. Although this

might be a gross exaggeration (given today's advanced encryption technology), it has become a real concern for pharmaceutical organizations. Beyond the issue of security, there is also the issue of functionality. In most instances, processing performance is hampered by programs and algorithms run over the Internet. Although these concerns can be addressed with necessary investments in telecommunications, it also provides a good excuse for pharmaceutical companies to simply purchase informatics software outright, and then run these programs on-site with mainframe processors. In summary, the ASP model offers a number of advantages for research organizations throughout the world. However, real (and perceived) privacy and performance concerns have prevented these vendors from making substantial inroads into the large pharmaceutical market. By contrast, this approach appears quite popular with academia and small biotechnology companies and organizations that naturally benefit from this business model's predictable, low-cost pricing structure.

5.2 Software Licenses (and Maintenance Fees)

To overcome the issues of security and performance, many informatics vendors have sold their software under perpetual license agreements. This common model allows for informatics applications to be installed on-site, facilitating data processing behind the four walls of the pharmaceutical company. Furthermore, to enhance the performance of these more advanced programs, many clients choose to invest in high-end mainframes to accelerate data output.
The obvious downside to the one-time license model is its unpredictability, with the majority of license revenues recognized up front. To minimize this effect, a number of vendors have charged clients an annually renewable maintenance fee, which typically ranges from 15% to 25% of the license revenue. In addition, consulting and data integration services (billed as services are provided) can also enhance the predictability of revenues.

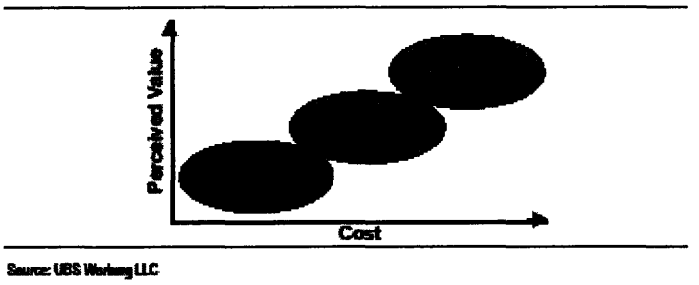


Figure 14: Perceived Value Versus Cost

5.3 Research Collaborations

In an effort to move beyond the role of just a software vendor, a number of informatics companies have combined their technology with wet lab services. We believe this bundled approach will prove popular with capacity-strained pharmaceutical companies looking to outsource a component of the drug discovery process. Furthermore, this business model not only moves informatics companies up the food chain in terms of perceived value (see Figure 14), but it also provides for closer relationships with pharmaceutical and biotechnology clients. Given the recent rollout of bioinformatics collaborations, it is difficult to describe the typical business model. But generally speaking, most contracts include a combination of fixed and incentive (milestone) payments. For example, LION Bioscience is working with Bayer to identify potential drug targets and genetic markers for disease susceptibility. As the company reaches predefined milestones, it will be awarded incentive payments (in addition to pro-rata fees). Most important, these collaborations allow for royalties on all drugs produced by these efforts, providing a potential stream of future cash flow. In many respects, the continued rollout of this outsourced offering will reposition many industry participants as *in silico* drug discovery ventures rather than software developers.

6 Commercialising the "Regulatory Sequence Analysis Package"

This section addresses the issues that need to be considered for commercialising the work produced from this thesis. In particular, this section will provide an overview of the TABS algorithm, the value that it adds for its customers, the ways in which it can be commercialised and the various issues that need to be considered in that process. Regulatory

Sequence Analysis falls under the realm of "functional genomics" in the value chain of drug discovery.

6.1 TABS – Teiresias-based Algorithm for identification of Binding Sites

This thesis led to the development of a software called TABS that enables identification of regulatory signatures in DNA sequences that control expression of genes. The program takes as input a set of upstream regions of "hypothetically" co-regulated genes, and performs pattern discovery analysis to identify motif signatures that are shared among those upstream regions and are possible involved in gene regulation. See Figure 15.

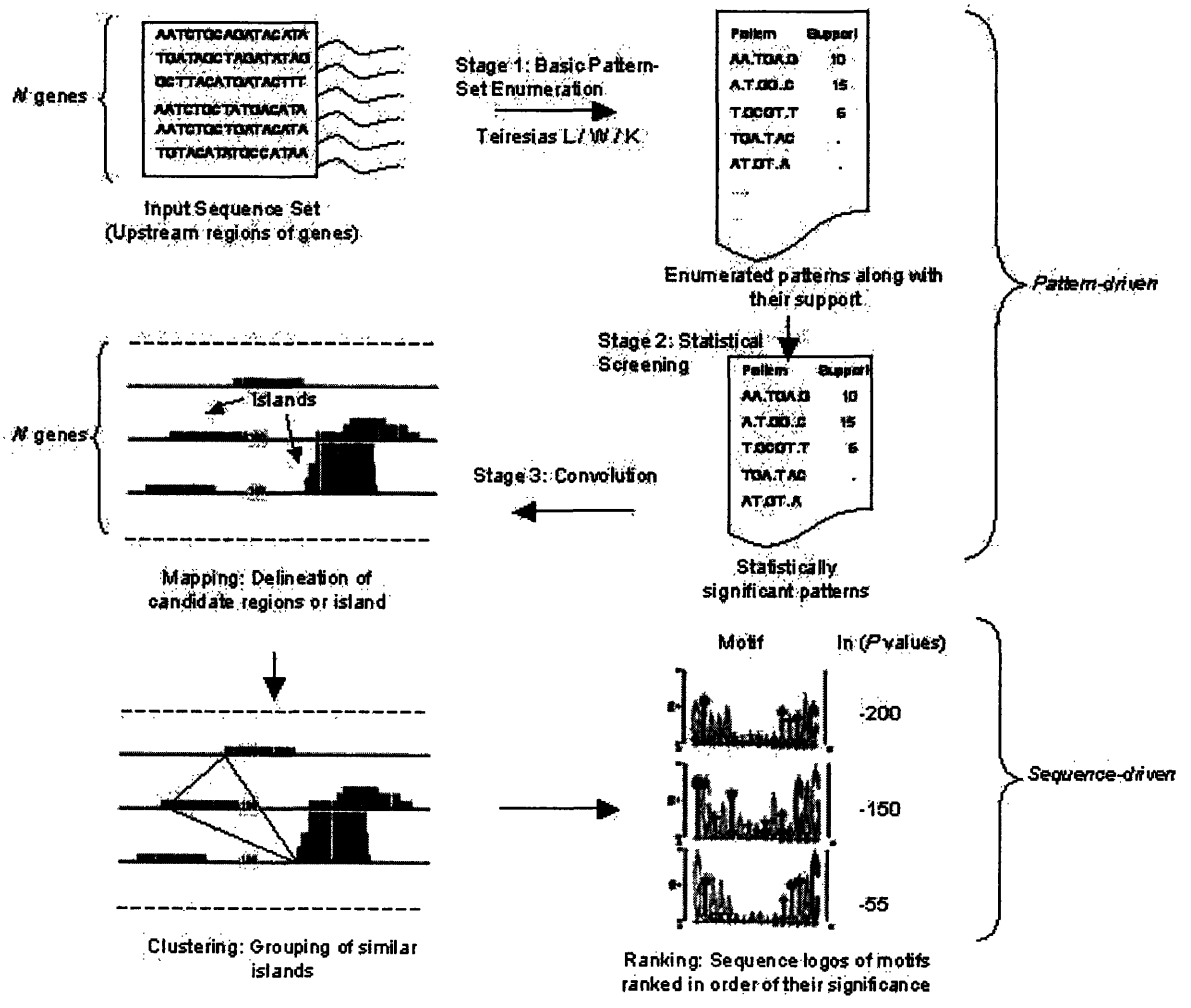


Figure 15: Overview of TABS Algorithm

6.2 Enhancing the knowledge derived form Gene Expression Analysis

Microarray data produces gene expression information that can be analysed to understand which genes are co-expressed, and which genes are implicated in a certain disease/ in a certain pathway. The regulatory sequence analysis package can be used to increase the confidence in the results from such an analysis.

The software searches for conserved regulatory sequences upstream of genes that are clustered based on their different expression profiles in response to external stimuli over time or across different tissue types. If these genes indeed share a common regulatory mechanism, then their upstream regions would be expected to contain a common binding site. The binding site or the discovered motif can throw some light on the nature of protein that regulates the genes and hence helps elucidate the regulatory network. Thus the end-customers of this product are all the biologists conducting drug discovery research in academic and commercial laboratories and want to understand the mechanism behind the various biological processes. See Figure 16.

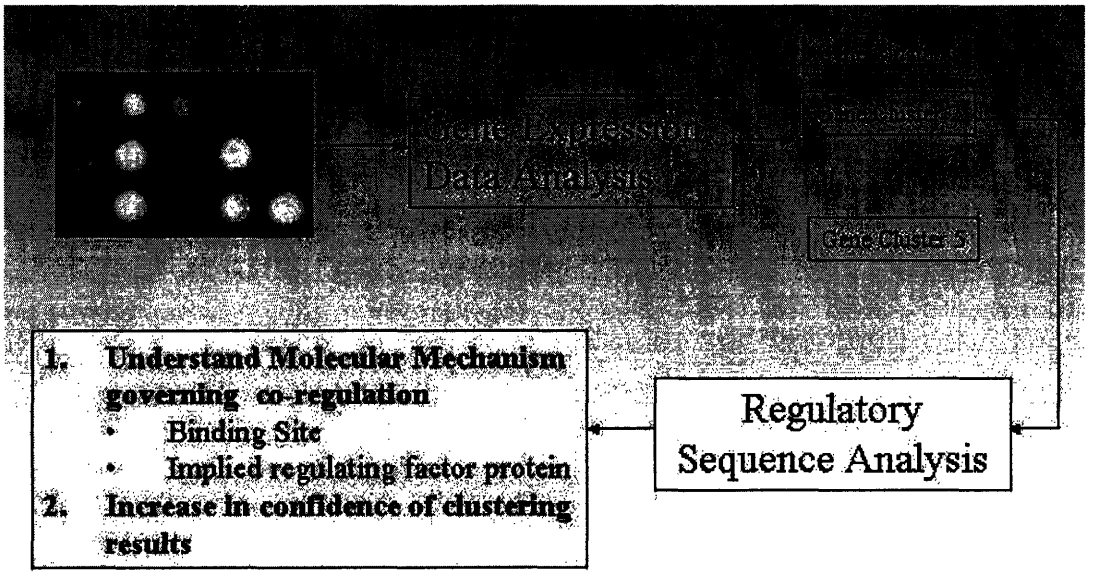


Figure 16: Integrating regulatory sequence analysis with microarray data analysis platform

6.3 Commercialising TABS

A first approach would be to attempt to commercialise this software as a complete tool by itself. The product being niche, however, this approach would have various downsides. Firstly, there is free-for-academic-use publicly available software that competes directly with

TABS. Examples of this are numerous: Consensus, dyad analysis, PromoterInspector, AlignACE, etc. Given that academic use constitutes a significant chunk of our customer base, it will be hard to make a profitable case. However, there is tremendous value in integrating the software with an existing functional genomics toolbox to deliver a complete solution at the desk of a researcher. This would also enable easier and faster penetration of the market through the already existing sales force of other established companies which could act as partners.

According to Frost and Sullivan, the entire gene expression market size was about $60M in 2003 and is growing at 10% an year (Source: Frost and Sullivan). The market is highly fragmented and several small analysis software companies as well as larger players compete in this market. Companies such as Affymetrix and Agilent sell bundled software with their gene-chip toolkits, thus penetrating a larger segment of the market. SpotFire, Inc is one of the leading players in this market with a revenue share of approximately $10M. Selling a license to any of these companies would be an attractive proposition.

TABS uses Teiresias®, copyright product of IBM. Before licensing TABS to any third party, commercial rights from IBM would need to be purchased. IBM's Life Sciences group already has a functional genomics product on market that includes a gene expression analysis tool (Genes@Work), Teiresias® and a sequence database. One possible approach would be to bundle TABS with this product and demand some royalty on sales from IBM. Since the basic software platform is common, integration issues would be minimal in this case. However, given that the market penetration of IBM's product is very small, this might not be a very attractive option.

6.4 Business Model

6.4.1 Integrate the product with a larger technology

Based on the previous discussion an attractive business model would be to license the product to a third-part player which is a market leader, such as SpotFire, Affymetrix or Rosetta. The official figures of revenues from the functional genomics toolbox is difficult to determine from the 10Ks of these companies (SpotFire is a private company, anyway) since these companies derive revenues across a range of products and services. However, we can estimate the yearly cash inflow by making some assumptions. For instance, SpotFire has annual revenues of $20M based on investor reports. SpotFire's revenues are primarily

derived from the sales of its product DecisionSite®, an enterprise software solution that provides visualization and analytical tools. 57% of its revenues come from the biotech/pharmaceutical sector. It sells a functional genomics and lead discovery add-on package to these companies. Table 4 shows prices of the base-software and each add-on. The number of users who use the functional genomics package has been estimated based on the total revenues and product price. Given that the functional genomics tool has 4 utilities (Visualization, PCA, k-means clustering and hierarchical clustering) for which a charge of $2000 per user/year is made, each utility has a worth of about $500. This would be a reasonable benchmark to use while pricing the TABS utility (Note: Of course, a better way would be to estimate the "value" that the software brings to SpotFire and its customers. However, in the absence of any direct comparables we are proposing a simple approach to estimating the value that we can demand). With a user base of 912 (Table 4), this implies an yearly revenue of $500x912=$456K for the company from the TABS utility. Assuming a 10% royalty back, we can make about $45K per year based on this simple model. A part of this would go back to IBM as royalty for using Teiresias®.

The primary upfront initial investment needed to start the company would go towards obtaining rights from MIT (since this IP belongs to MIT). Other steps involving costs would be to reach out and sell the product to potential third-part customers and also product development.

Base Price	Browser	$3,500 per user/yr
	Server	$5,500 per user/yr
	Total	$9,000
Add-ons	Functional Genomics Package Add-on	$2,000 per user/yr
	Lead Discovery Package Add-on	$1,500 per user/yr
Total Price		$12,500 per user/yr
Total Revenue	Total	$20,000,000 per year
	Biotech	$11,400,000 per year
Implies user base		912
Price of TABS feature		$500 per user/year
Revenues from TABS		$456,000 per year

Table 4: Licensing to SpotFire

6.4.2 Generate IP related to metabolic pathways and regulatory network

An alternate business model would be to sell "content" in the form of information related to metabolic pathways and regulatory networks related to specific diseases which drug-discovery companies would be willing to buy. Starting from public microarray gene

expression databases, one could use TABS to analyse and search for regulatory signatures. Together, this information along with wet-bench experiments for validation, can be used to reconstruct pathways. This business model could lead to higher revenues than the previous one, since it is possible to demand higher for "content" databases. However, the operating cost associated will also be larger (maintaining laboratories, paying skilled labour, etc).

6.5 Reaching out to the customer

A very important part of the selling process would be to reach out to "sell" the product to potential customers and convince them about the utility of the product. Since the end-customers are scientists and people who work in the academic laboratories, a scientific publication in a renowned journal could serve as a key marketing tool. Once the credibility is established, it will be a lot easier to approach the customers and demonstrate live the value of the software.

6.6 Recommendation

Based on the above analysis, we conclude there are several options to commercialise TABS. The more attractive options are those where the technology can be made a part of a larger platform and bundled with a pre-existing and established software product of a pre-established player such as Rosetta Inpharmatics, SpotFire or Affymetrix. An alternate model would be operate as a "content" company that sells intellectual property related to specific pathways that can serve to find targets for particular diseases. We believe, given the trends in the industry, that this model would be very attractive in the long term.