ANALYSIS OF DNA MOTIFS IN THE HUMAN GENOME

by

YUPU LIANG

A dissertation submitted to the Graduate Faculty in Computer Science in partial fulfillment of the requirements for the degree of Doctor of Philosophy, The City University of New York

2014

This manuscript has been read and accepted for the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

Dina Sokol

Date — Chair of Examining Committee

Theodore Brown

Date — Executive Officer

Susan Imberman

Saad Mneimneh

Sarah Zelikovitz

Terry Gaasterland

Supervision Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

Analysis of DNA motifs in the Human Genome

by

Yupu Liang

Advisor: Professor Dina Sokol

DNA motifs include repeat elements, promoter elements and gene regulator elements, and play a critical role in the human genome. This thesis describes a genome-wide computational study on two groups of motifs: tandem repeats and core promoter elements.

Tandem repeats in DNA sequences are extremely relevant in biological phenomena and diagnostic tools. Computational programs that discover tandem repeats generate a huge volume of data, which can be difficult to decipher without further organization. A new method is presented here to organize and rank detected tandem repeats through clustering and classification. Our work presents multiple ways of expressing tandem repeats using the n-gram model with different clustering distance measures. Analysis of the clusters for the tandem repeats in the human genome shows that the method yields a well-defined grouping in which similarity among repeats is apparent. Our new, alignment-free method facilitates the analysis of the myriad of tandem repeats replete in the human genome. We believe that this work

will lead to new discoveries on the roles, origins, and significance of tandem repeats.

As with tandem repeats, promoter sequences of genes contain binding sites for proteins that play critical roles in mediating expression levels. Promoter region binding proteins and their co-factors influence timing and context of transcription. Despite the critical regulatory role of these non-coding sequences, computational methods to identify and predict DNA binding sites are extremely limited. The work reported here analyzes the relative occurrence of core promoter elements (CPEs) in and around transcription start sites. We found that out of all the data sets 49%-63% upstream regions have either TATA box or DPE elements. Our results suggest the possibility of predicting transcription start sites through combining CPEs signals with other promoter signals such as CpG islands and clusters of specific transcription binding sites.

Acknowledgements

Many thanks to my committee members, colleagues, and family for your support in this work.

The research was supported by: National Science Foundation Grant DBI 0542751 and PSC-CUNY Research Award 63343-0041

Table of Contents

List of Figures

Abbrevations

Abbreviations

A	**A**denine
T	**T**hymine
C	**C**ytosine
G	**G**uanine
U	**U**racil
DNA	**D**eoxyribo**n**ucleic **a**cid
RNA	**R**ibo**n**ucleic **a**cid
CPE	**C**ore **P**romoter **E**lement
TSS	**T**ranscription **S**tart **S**ite
PWM	**P**osition **W**eight **M**atrix

To my parents, my husband and my son

Part I

Background and Introduction

Chapter 1

Basics of Molecular Biology

The basic concepts and notation of molecular biology relevant to the computational identification and analysis of regulatory motifs are reviewed here.

1.1 Concepts

DNA is the hereditary material in humans and most other organisms. It was first isolated by Miescher in 1868 and its double helix structure was solved by Crick and Watson in 1953, based on X-ray diffraction data from Franklin and Wilkins. Most DNA is located in the cell nucleus, but a small amount of DNA can also be found in the mitochondria (mtDNA). DNA is made of chemical building blocks called nucleotides, as shown in figure 1.1. Nucleotides are made of three parts: a phosphate group, a sugar group, and one of four types of nitrogen bases: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). To form a strand of DNA, nucleotides are linked into chains, with the phosphate and sugar groups alternating. The order, or sequence, of these bases determines which biological instructions are contained in a strand of DNA. A **gene** is a DNA sequence that contains

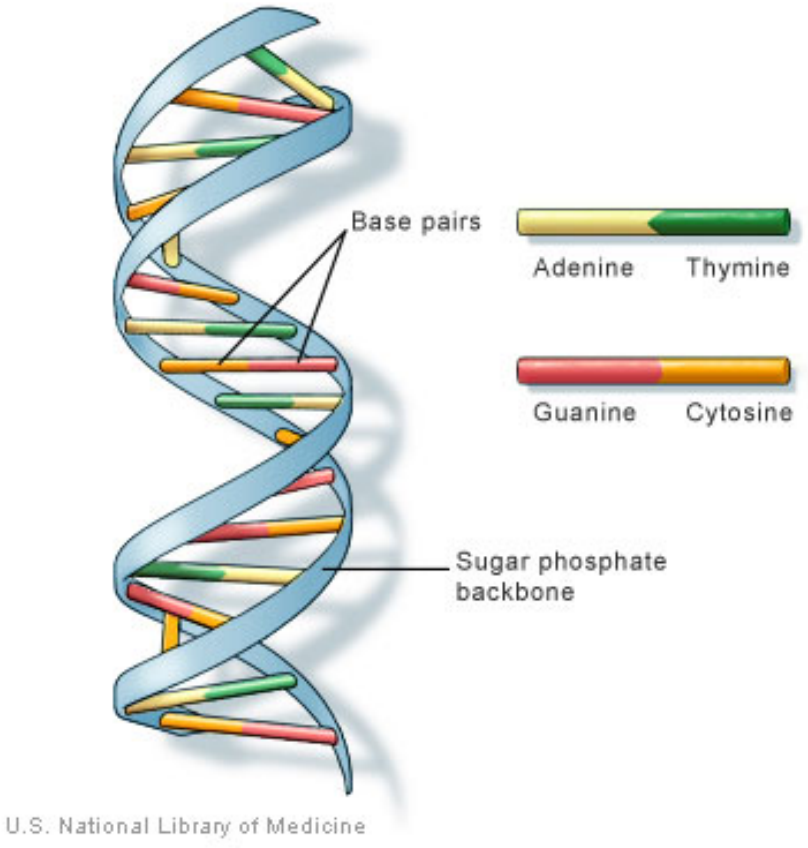

Figure 1.1: DNA structure

instructions to make a protein (through RNA as described shortly). The size of a gene may vary greatly, ranging from about 1,000 bases to 1 million bases in human. A **chromosome** is made up of DNA tightly coiled many times around proteins called histones that support its structure. The complete human genome contains about 3 billion bases and about 20,000 genes on 23 pairs of chromosomes. This is shown in figure 1.2.

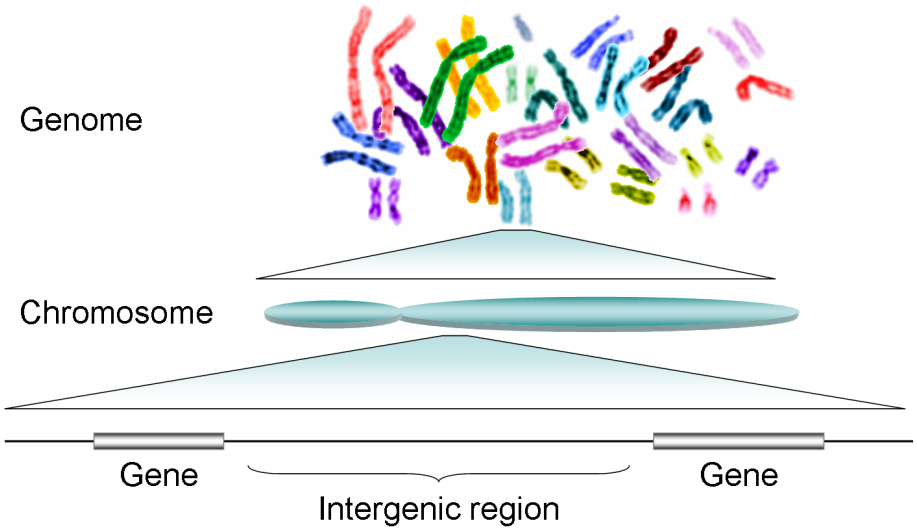

Figure 1.2: Genome, Chromosome and Genes [6]

RNA is a chemical analogous to a single strand of DNA. In RNA, the nitrogen base U is substituted for T in the DNA. RNA is used as the genetic template to make proteins.

Proteins are chains of small molecules, called amino acids, which consist of a central carbon atom connected to an amino group, a carboxyl group, and a side chain. In nature, there are several known amino acids, but only twenty of them serve as the standard building blocks of proteins.

The Central dogma, shown in figure 1.3, is the backbone of molecular biology. Molecular biology describes how DNA information is used to make proteins. First, DNA is used as the template to transcribe genetic information into messenger RNA (mRNA). This step is

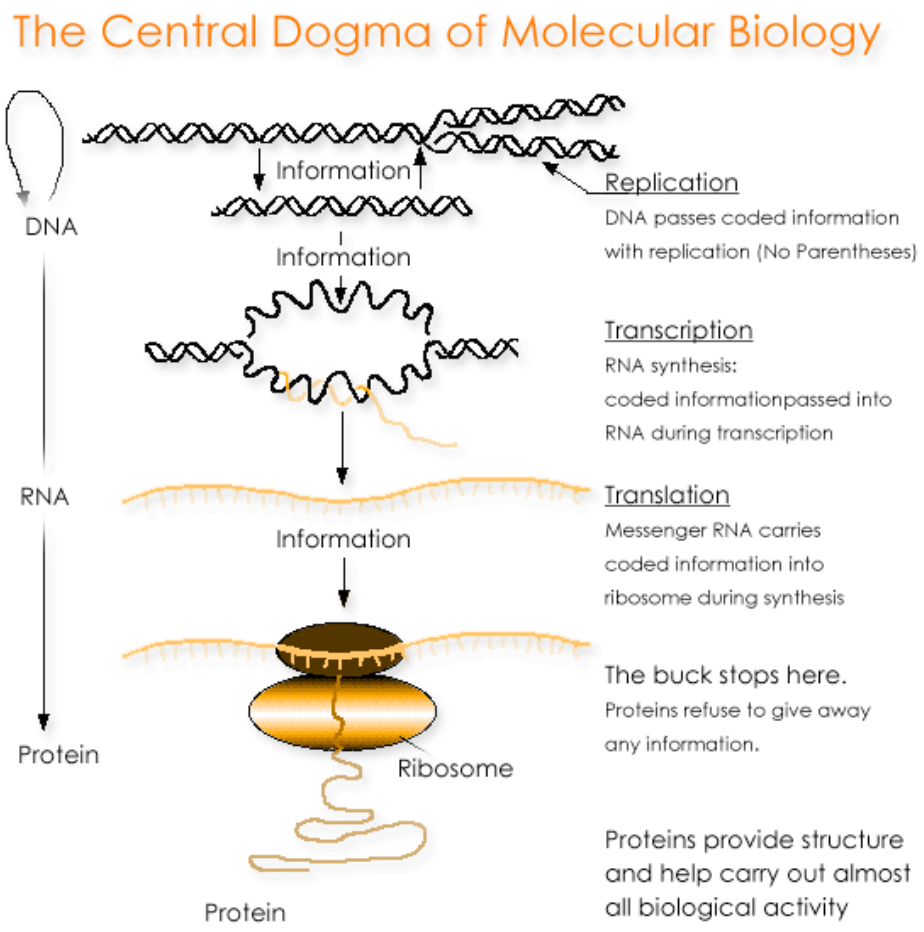

Figure 1.3: Central Dogma [3]

called transcription. Next, the information contained in the mRNA is translated into amino acids, which are the building blocks of proteins. Thus, this second step is called translation.

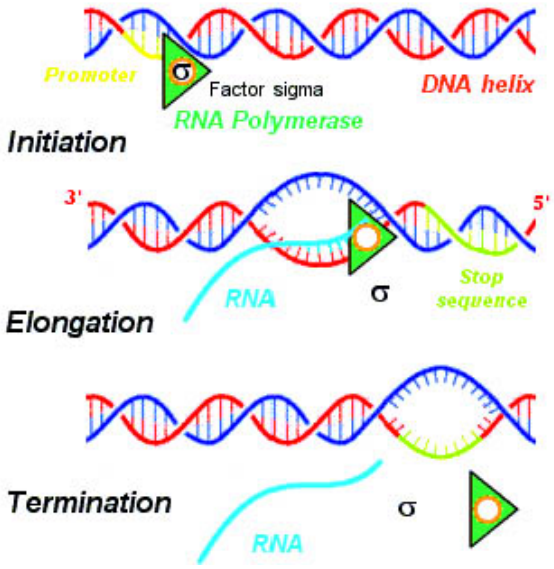

Figure 1.4: Transcription
[2]

Transcription, illustrated in more detail in figure 10.2, is the general process of copying genomic DNA into mRNA. It is part of the regulation procedure that decides which gene is going to be expressed and the degree to which, it is going be expressed. Transcription is shaped by the interaction between transcription factors (proteins) and regulatory elements (DNA binding sites).

The Promoter is the best studied regulatory element and its function is to mediate and control initiation of transcription of the gene. The Promoter is located immediately upstream of the regulated gene, as shown in figure 1.5.

Repetitive DNA occurs in two forms figure 1.6: genome-wide repeats, whose individual repeat units are distributed around the genome in seemly random fashion, and tandem repeats, whose repeat units recur next to each other in an array. Four types of repeats

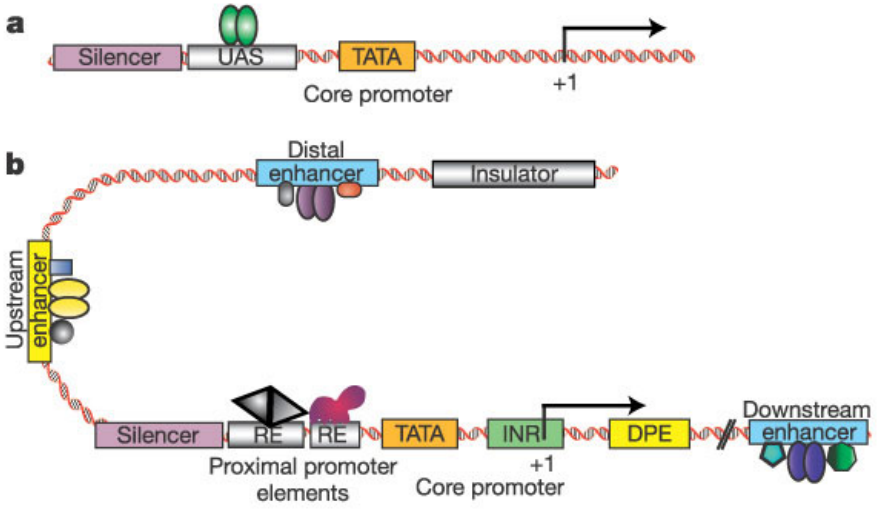

Figure 1.5: Promoter [1]

dominate the human genome: SINEs, LINEs, LTR elements and transposons. Altogether, genome-wide repeats make up about 44% of the human genome[54, 86]. Tandem repeats are also termed satellite DNA because DNA fragments containing tenderly repeated sequences form 'satellite' bands when genomic DNA is fractionated by density gradient centrifugation. Although they do not appear in satellite bands on density gradients, two other types of DNA tandem repeats are also classified as 'satellite' DNA: minisatellites and microsatellites. A minisatellite repeated unit is a short series of bases on the order of 100bp in length, whereas the microsatellite's unit is usually a few bases.

1.2 Experimental Techniques

DNA sequencing refers to methods for determining the order of the nucleotide bases in DNA. Sanger sequencing, or the chain-termination method, is the most famous method because of its efficiency and reliability. As sequencing methods can only generate a few

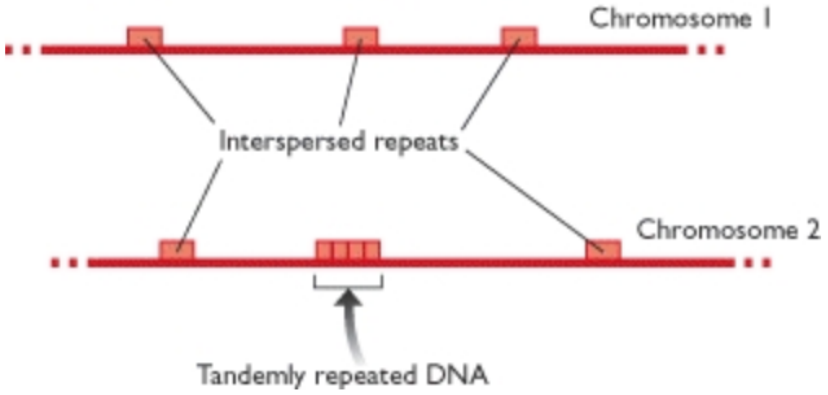

Figure 1.6: Two types of Repetitive DNA [22]

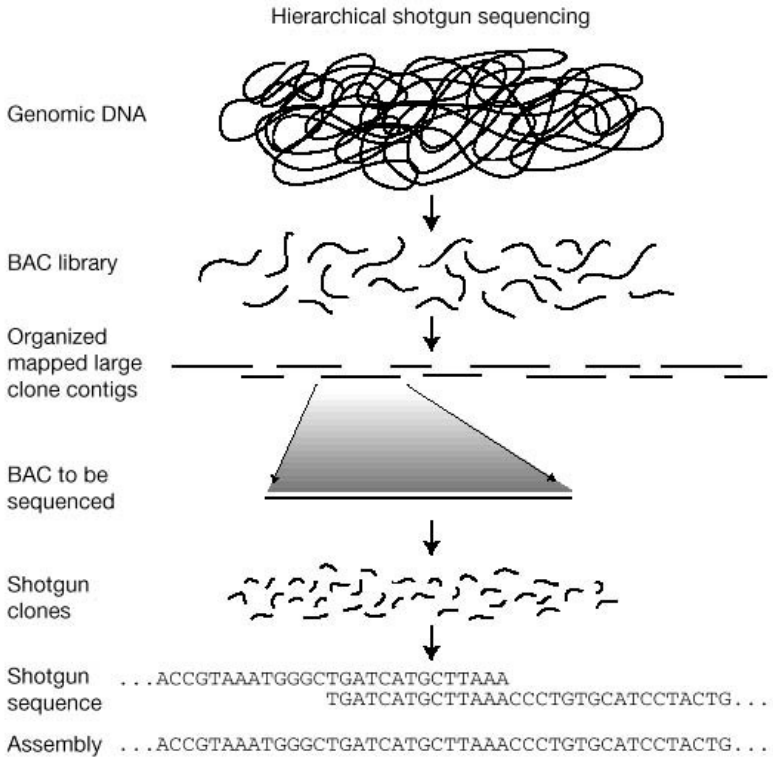

Figure 1.7: Sequence Assembly [7]

hundred nucleotides, a common approach to sequencing long DNA pieces (large-scale sequencing), such as a whole chromosome (10,000 to 1,000,000,000 nucleotides) involves a divide-and-conquer technique known as shotgun sequencing. In shotgun sequencing, the long DNA is first cut into smaller pieces (500-1000 bp) so that they can be directly sequenced. After the small pieces are sequenced, they are combined into a bigger piece that could represent the original DNA. The second step, which is often called sequence assembly is shown in figure 1.7. Several new sequencing technologies have emerged recently. The intention here is to decrease the sequencing cost by parallelizing the sequencing process. One such method, which produces thousands of millions of sequences simultaneously, is 454 sequencing shown in figure 1.8. The tradeoff with this approach is that it can only work with much smaller sequences than the sequential methods can. Shorter reads pose difficulties at the assembly stage [20, 78].

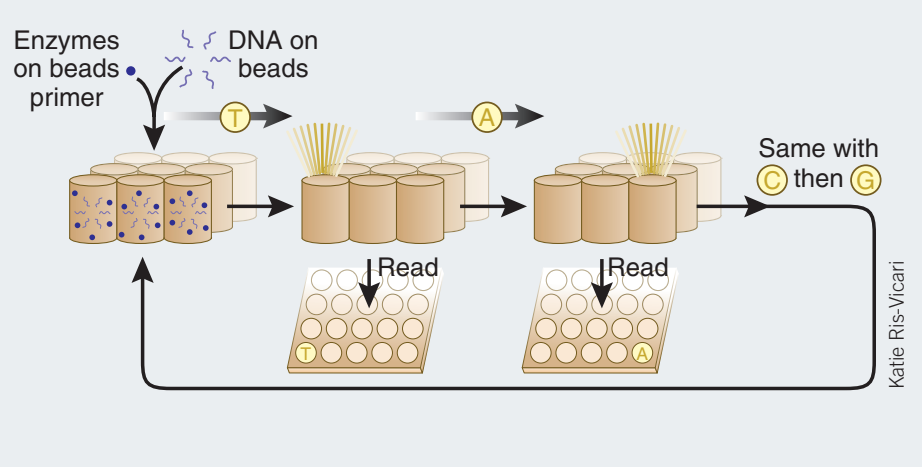

Figure 1.8: PyroSequencing [48]

CHIP-chip is a technique that combines chromatin immunoprecipitation ("ChIP") with microarray technology ("chip") [13, 12]. Like regular ChIP [26], ChIP-chip is used to investigate interactions between proteins and DNA in vivo. Whole-genome analysis can be

performed to determine the locations of binding sites for almost any protein of interest [72] (figure 1.9). As the name of the technique suggests, such proteins are generally those operating in the context of chromatin. The most prominent representatives of this class are transcription factors and replication-related proteins. The goal of ChIP-chip is to localize protein binding sites that may help identify functional elements in the genome. For example, in the case of a transcription factor, one can determine its transcription factor binding sites throughout the genome. Other proteins allow the identification of promoter regions, enhancers, repressors and silencing elements, insulators, boundary elements, and sequences that control DNA replication.

1.3 Genome Projects

Genome projects are scientific endeavors that aim to determine the complete genome sequence of an organism and to annotate protein-coding genes and other important genome-encoded functional elements. The genome sequence of an organism includes the collective DNA sequences of each chromosome in the organism. The human genome includes 22 pairs of autosomes and 2 sex chromosomes, a complete re-sequencing a person's genome will involve 'reading' 46 separate chromosome sequences.

Many organisms have genome projects that have either been completed or will be completed shortly. A number of salient genomes include:

- Humans, Homo sapiens
- The Rice Genome
- Palaeo-Eskimo, an ancient-human

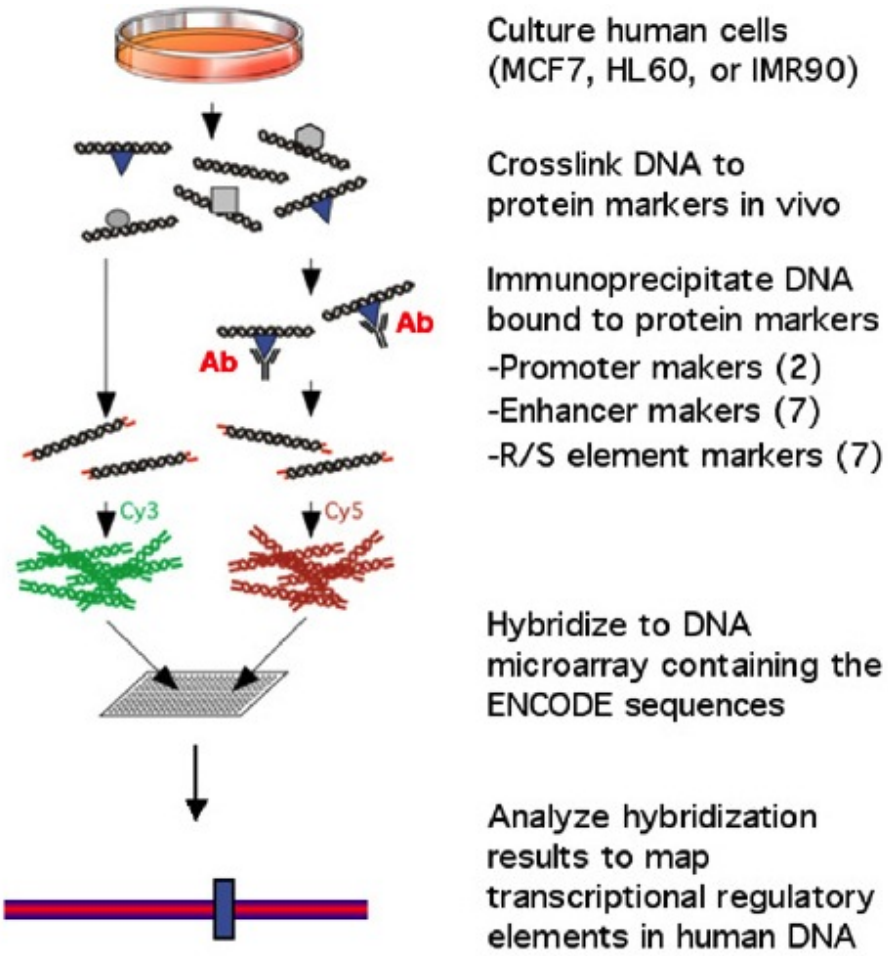

Figure 1.9: Genome Wide CHIP-chip analysis

- Neanderthal, ”Homo neanderthalensis”
- Common Chimpanzee Pan troglodytes
- Domestic Cow
- Honey Bee Genome Sequencing Consortium
- Horse genome
- Human microbiome project
- Canis lupus familiaris (dog)
- Fugu genome

Completed in 2001 [54, 86] the Human Genome Project (HGP) was a 13-year project coordinated by the U.S. Department of Energy and the National Institutes of Health. During the early years of the HGP, the Wellcome Trust (U.K.) became a major partner. Additional contributions came from Japan, France, Germany, China, and others. Project goals were to:

1. Determine the sequences of the 3 billion chemical base pairs that make up human DNA
2. Identify the approximately 20,000-25,000 genes in human DNA
3. Store this information in databases
4. Improve tools for data analysis

Though the Human Genome Project (HGP) is finished, analyses of the data will continue for many years. In addition to predicting where DNA regions encoding proteins are located, a major effort will be locating other DNA elements such as repeat elements and transcription regulatory elements which are equally important and pose a computational and experimental challenge. The ENCODE pilot Projects [19, 28] comprise a major undertaking to examine transcriptional regulation systematically. Data gathered through ENCODE has already yielded new understanding about transcription start sites, including their relationship to specific regulatory sequences and features of chromatin accessibility and histone modification.

Chapter 2

Basics of String and Graph Algorithms

There is a natural mapping between the biological sequence and the string data structure in computer science. We review some concepts, and basic algorithms that have been applied to bioinformatics.

2.1 Definitions

Biological sequence data can be represented by strings, eġ; DNA, as a string of ATGCs. Here, we give the formal definition of a string.

Definition 2.1.1. [1]*An alphabet Σ is a finite nonempty set of symbols. The elements of an alphabet are called characters, letters, or symbols. A string s over an alphabet Σ is a concatenation of symbols from Σ. The length of a string s is the number of symbols in s; it is denoted by $|s|$.*

The empty sting λ denotes the string of length 0. Σ^n is the set of all strings of length n over the alphabet Σ. $\Sigma^* = \bigcup_{i \geq 0} \Sigma^i$ is the set of all strings over Σ.

Definition 2.1.2. *Let Σ be an alphabet, $s = s_1 \cdots s_n,\ s_1, \cdots, s_n \in \Sigma$, be a string. For all $i, j \in 1, \cdots, n, i < j$, $s[i, j]$ is the substring $s_i \cdots s_j$. Furthermore, $s[i]$ is the the i-th symbol of s, s_i.*

The Tree data structure is another widely used representation in bioinformatics.

Definition 2.1.3. *An (undirected) graph G is a pair G=(V, E), where V is a finite set of vertices and E $\subseteq \{(x, y)|x, y \in V and\ x \neq y\}$ is a set of edges. The degree of a vertex x is the number of edges incident to x. A path in G is a sequence of vertices $P = x_1, x_2, \cdots x_m, x_i \in V, \forall i \in 1, \cdots, m$ such that $\{x_i, x_{i+1}\} \in$ E holds for all $i \in 1, \cdots, m-1$. A path is called a simple cycle if $x_1 = x_m$ and $x_1, x_2, \cdots, x_{m-1}$ are pairwise different.*

Definition 2.1.4. *Let T=(V, E) be a graph. The graph T is a tree, if it is connected and does not contain any simple cycle. The vertices of degree 1 in a tree are called leaves; the vertices of degree ≥ 2 are called inner vertices. A tree may have a specially marked vertex that is called the root. In this case, the tree is a rooted tree.*

2.2 String Matching Algorithms

Bioinformatic algorithms are normally not exact algorithms as they need to allow for a certain amount of error to accommodate experimental error or fuzzy biological definitions. Yet, several basic string algorithms and data structures have been widely used in bioinformatics and lead to either direct solutions or functions of more complex solutions. Here, we review some string matching algorithms that are related to this thesis work.

[1]The definitions of this chapter are adapted from [21]

The *string matching problem* is probably the most elementary problem when dealing with sequence data or a string. The problem is that of finding a substring or pattern in a given (usually very long) string. This problem arises in many non-biological applications as well, iė; in text editors or search engines. An important bioinformatics application is the search for a known gene in newly sequenced DNA.

Definition 2.2.1. *Let Σ be an arbitrary alphabet. The string matching problem is: Input: Two strings $p = p_1 \dots p_m$ and $t = t_1 \cdots t_n$ over Σ. Output: The set of all positions in the text t, where an occurrence of the pattern p as a substring starts, i.e. a set $I \subseteq 1, \cdots, n - m + 1$ of indices, such that $i \in I$ if and only if $t_i \cdots t_{i+m-1} = p$.*

Algorithm 1 Naive string matching algorithm

Input: a pattern $p = p_1 \cdots p_m$ and a text $t = t_1 \cdots t_n$
$I = \emptyset$
for $j = 0$ to $n - m$ **do**
  $i = 1$
  while $p_i = t_{j+i}$ and $i \leq m$ **do**
    $i = i + 1$
  end while
  if $i = m + 1$ **then**
    $I = I \bigcup j + 1$
  end if
end for
Output: The set I of positions where an occurrence of p begins in t

2.2.1 Naive Approach

The naive Algorithm 1 uses a sliding window the size of pattern p over text t, and tests for each position of the window, whether there is a perfect matching between the substring inside the window and p. Let $|p| = m$ and $|t| = n$. In the worst case, this algorithm needs m comparisons for each value of i, with an overall running time of $O(m(n - m)) = O(mn)$.

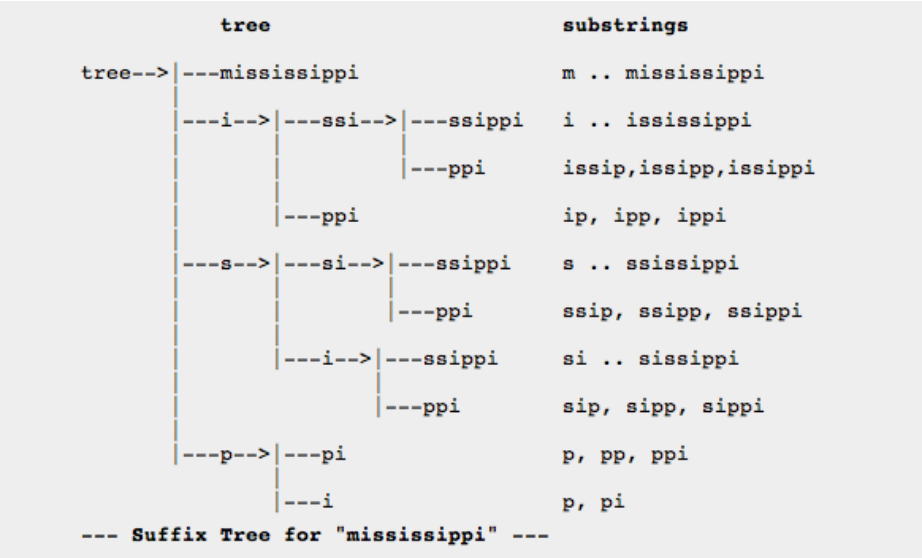

Figure 2.1: Suffix Tree [4]

2.2.2 Suffix Tree and Suffix Array

In string matching, a pattern occurs in the text if and only if it is the prefix of a suffix of the text. The suffix tree is a data structure that indexes the suffixes of the input text. The suffix tree in figure 2.1 was first applied to string matching problem by Aho, Hopcroft and Ullman [10]. This method preprocesses the text, with a significant speed increase when many different patterns will be matched to the same text.

Definition 2.2.2. *Let $t = t_1 \cdots t_n \in \Sigma^n$ be the text. A directed tree $T_t = (V, E)$ with root r is called a suffix tree of t if it satisfies the following conditions:*

1. *The tree has exactly n leaves which are labeled $1, \cdots, n$.*
2. *Every internal vertex of T_t has at least two children.*
3. *Each edge of the tree is labeled with a substring of t.*

4. *The outgoing edges from an inner vertex to its children are labeled with pairwise different symbols.*

5. *The path from the root to leaf i is labeled $t_i \cdots t_n$.*

The original string is padded with a terminal symbol \$, which is not part of the alphabet, to ensure that no suffix is a prefix of another substring.

Weiner [88] and McCreight [59] were the first to show that a compact suffix tree for a given string $t = t_1 \cdots t_n$ can be constructed in linear time. An on-line algorithm was later designed by Ukkonen [85].

Once the suffix tree is constructed, the string matching problem can be solved by traversing the path in the tree that starts at the root and is labeled by the given pattern. All the leaves in the subtree rooted with this path correspond to positions in the text where the pattern starts.

Constructing the suffix tree is $O(n \log |\Sigma|)$ time for a text of length n and searching a given pattern is $O(m \log |\Sigma| + k)$ where m is the length of the pattern and k is the number of occurrences.

A suffix array is an array describing the lexicographical order of all suffixes of a given string. This data structure can be constructed in linear time and can solve the string matching problem in $O(m \log n + k)$ time [57], where k is the number of occurrences of the pattern. Recently, three new different algorithms were introduced to directly construct a suffix array in linear time [49, 44, 50].

The suffix tree and suffix array can be modified to solve many other string problems efficiently such as finding a substring in a set of texts, longest common substring, overlap of strings, and repeats in strings. The suffix tree or suffix array are also often a component of more complicated algorithms.

Chapter 3

Pairwise Sequence Alignment Algorithms

Sequence alignment is a traditional bioinformatics task that has multiple applications, such as comparing the same gene between different species, searching for a novel DNA sequence against all known genes, or identifying functionally important amino acids of protein sequences. In theory, we can align the sequences using string algorithms from the previous chapter, but, this does not work in practice because of the following reasons:

- Sequences obtained from experiments are subject to measurement errors.
- Sequences are prone to small changes (mutation) between individuals (person A vs. person B) or for the same object at different time (E.coli strain A from two years ago vs. the same strain today)
- Genes that serve the same function have different sequences in different species.

3.1 Alignment and Scoring function

Definition 3.1.1. *Let $s = s_1 \cdots s_m$ and $t = t_1 \cdots t_n$ be two strings over an alphabet Σ. Let $- \notin \Sigma$ be a gap symbol and let $\Sigma' = \Sigma \cup -$. Let $h : (\Sigma')^* \to \Sigma^*$ be the mapping $h(a) = a$ for all $a \in \Sigma$, and $h(-) = \lambda$.*

An alignment of s and t is a pair (s', t') of strings of length $l \geq max\{m, n\}$ over the alphabet Σ', such that the following conditions hold:

1. $|s'| = |t'| \geq max\{|s|, |t|\}$,
2. $h(s') = s$,
3. $h(t') = t$ *and*
4. *there is no i such that s'_i =t'_i= gap*

Example 3.1.1. *Let $s = GGGGATTTT$ and $t = GGGGTTAT$. A possible alignment of s and t is:*

```
s' = GGGGATTTT
t' = GGGG-TTAT
```

For the previous example, the alignment can be viewed as a 2 by l matrix with four kinds of column-wise alignments: insertion, deletion, match and mismatch. The alignment can be scored by summing up over the columns:

$$M(s', t') = \sum_{i=1}^{l} M(s'_i, t'_i). \tag{3.1.1}$$

Function M is called the score function. Given the score function, the sequence alignment problem can be solved by finding s' and t' that $M(s', t')$ is optimized. There are many

ways to define the score function. A simple score function could be defined as $m(a, a) = 1$, $m(a, b) = -1$ if $a \neq b$. When dealing with biological sequences, several scoring methods have been established and they are often presented as score matrices. PAM matrices and BLOSUM matrices are score methods that incorporate evolution data into account.

In practice, there are different criteria for the final alignment: global alignment vs local alignments.

1. A global sequence alignment optimizes the alignment of the two entire sequences.

2. A local sequence alignment attempts to optimally align subsequences of the input sequences this allows arbitrary-length segments of each sequence to be aligned, with no penalty for the unaligned portions of the sequences.

Example 3.1.2. *Let $s = TGGTATTCC$ and let $t = TTATCCG$. A possible global alignment of s and t is:*

```
s'=TGGTATTCC-
t'=T--TAT-CCG
```

And a possible local alignment is

```
s' =TGGTATTCC--
t' =---TTAT-CCG
```

We achieve different alignment outcomes through applying different scoring functions.

3.1.1 Alignment between two sequences

The main issue with solving the sequence alignment problem is that in an optimal alignment, all its prefixes need to be optimal too, thus, one can solve the problem recursively through dynamic programming.

In order to use dynamic programming, an $(m \times n)$ matrix sim is labeled with rows as $s_1...s_m$ and columns as $t_1...t_n$. Each entry $sim(i, j)$ is the score of an optimal alignment of $s_1...s_i$ and $t_1...t_j$. Particularly, the entry $sim(m, n)$ gives the score of an optimal alignment of s and t. This matrix is often called a similarity matrix.

Obviously, an algorithm can be found through constructing the similarity matrix with running time $O(mn)$ and memory space $O(mn)$. The memory requirement can drop to $O(m + n)$ when applying a modification such as Hirschberg's algorithm [39].

Needleman-Wunsch algorithm

The NeedlemanWunsch algorithm [62] is an example of dynamic programming and was the first application of dynamic programming to biological sequence comparison.

There are two steps in this algorithms: construct the similarity matrix and trace back the matrix to find the optimal alignment.

Define g as the gap penalty function as $g = -2$:

The similarity matrix is constructed based on formula 3.1.2.

$$sim(s_1...s_i, t_1...t_j) = max \begin{cases} sim(s_1...s_{i-1}, t_1...t_j) + g \\ sim(s_1...s_i, t_1...t_{j-1}) + g \\ sim(s_1...s_{i-1}, t_1...tj - 1) + p(s_i, t_j) \end{cases} \tag{3.1.2}$$

For example, in order to align $GCCCTAGCG$ with $GCGCAATG$, A matrix will need to be populated, with trace back information(pointers) as in figure 3.1:

First, the similarity matrix is initialized by filling in the scores and pointers for the second row and second column. Traveling to the right in the second row corresponds to aligning the character in the first sequence along the top with a space(gap), rather than the first character of the sequence on the left. The gap penalty is -2, so each time this happens,

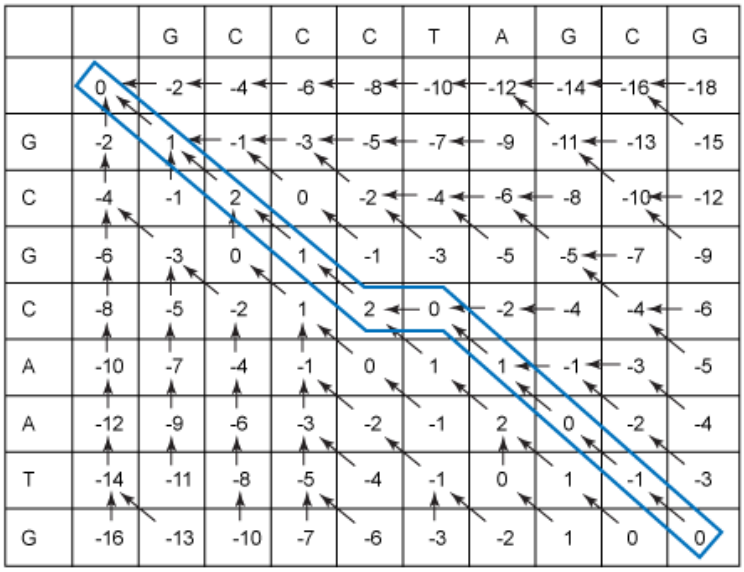

Figure 3.1: Similarity matrix for Needleman-Wunsch algorithm[5]

the score is -2 to the previous cell. The previous cell is the one to the left. Therefore this explains how 0, -2, -4, -6, ... sequence gets to be placed in the second row. Similarly, the second column gets initialized downward.

Next, the remaining elements of the matrix need to be filled by the maximum of the following three directions: from above $sim(s_1...s_{i-1}, t_1...t_j) + g$, from the left $sim(s_1...s_i, t_1...t_{j-1}) + g$, or from the above-left $sim(s_1...s_{i-1}, t_1...tj - 1) + p(s_i, t_j)$.

The final alignment for this example is:

```
s'=GCCCTAGCG
t'=GCGC-AATG
```

Smith-Waterman algorithm

The Smith-Waterman algorithm is a variation of the Needleman-Wunsch algorithm that was proposed by Temple Smith and Michael Waterman in 1981 [80]. The difference between

the two algorithms is how the score function $sim(s,t)$ is defined. And it is defined as follows:

Given g as the gap penalty function and $sim(s_i,-)=0$ $sim(-,t_j)=0$:

$$sim(s_1...s_i,t_1...t_j)=max\begin{cases} sim(s_1...s_{i-1},t_1...t_j)+g \\ sim(s_1...s_i,t_1...t_{j-1})+g \\ sim(s_1...s_{i-1},t_1...tj-1)+p(s_i,t_j) \\ 0 \end{cases} \tag{3.1.3}$$

When one compares the formula 3.1.3 with the formula 3.1.2, it is clear that the Smith-Waterman algorithm differs from the Needleman-Wunsch algorithm in the following three ways:

- In the initialization stage, the second row and second column are all filled with 0s regardless of gap penalty.
- In the matrix population stage, a 0 is placed in whenever a negative score occurs, and the pointer is added only for those cells that have positive scores.
- In the traceback stage, the tracing starts with the cell that has the highest score and works backward until a cell with a score of 0 is reached.

The basic idea of this modification is that if the penalty to extend the current alignment is too big(negative score), it is a better idea to start a new alignment (set it with zero).

The running time and memory requirements of the Smith-Waterman algorithm are the same as the Needleman-Wunsch algorithm.

3.1.2 Alignment between one sequence and a set of sequences

When one sequence needs to be aligned against a database that contains millions of sequences, polynomial algorithms are too slow. Several heuristic algorithms were proposed to address the efficiency problem in exchange for not guaranteeing the optimal solution.

BLAST

A commonly used heuristic algorithm is BLAST [11]. This program exists in many different implementations that are optimized for different tasks, for example, how it searches for similar sequences DNA or Protein. The main idea behind BLAST is as follows:

- BLAST looks for hits: similar subsequences between the query sequence and target sequences in the database of a given length w. Typical values for the w are $w = 11$ for DNA and $w = 3$ for proteins. Instead of requiring a perfect match, BLAST searches for non-gapped matches that exceed a certain similarity threshold. For example, PQG, PEG, PRG, PKG are all considered as hits for PQG.

- Each hit is filtered based on whether it is located within a certain distance d of another hit. Those failing this requirement are not considered in the subsequent step. The value d depends on the value w from step 1.

- BLAST then extends alignment from the paired hits of the previous step by adding further alignment in both directions until the similarity score does not increase any further. All the extended alignments that pass a certain threshold score are called a high scoring pair (HSP). The output of BLAST algorithms are all the HSPs.

BLAST is statistically less sensitive. However it is more efficient and it also takes the sequence composition of database into account and gives each HSP a measurement of statistical significance.

BLAT

BLAT [47] (short for BLAST-like alignment) is a modification of BLAST. It is designed to align mRNA/DNA sequences much faster (50 times - 500 times depending on the setting) and with more accurate results. BLAT's unique features include:

1. Instead of building an index of the query sequence as BLAST(hits) and then scanning through the database sequence, BLAT builds an index of all possible length w sub-sequences of the database then it scans through the query sequence. This is because the database only needs to be indexed once and the scanning of w hits is much faster than scanning through the database sequences.

2. BLAT is designed to align sequences of 95% and greater similarity of length 40 bases or more [47]. BLAT can trade sensitivity for speed: hits with an almost perfect match that far away from each other are not filtered out like in BLAST. Any hit whose alignment passes a certain threshold can trigger an extension of an alignment in BLAT.

3. When there are gaps left between aligned blocks in both the database and query sequence, the exact matching algorithm is performed on the gap in an attempt to find smaller alignable blocks that are missed from indexing. However BLAT will miss remote matches that are shorter than w and not near any aligned blocks.

4. Instead of reporting HSP as BLAST does, BLAT stitches together the extended

aligned regions into a larger alignment block. It gives the result in a more user-friendly format.

Chapter 4

Motif Finding Algorithms

In bioinformatics a motif is defined as a short sequence pattern that has some biological significance. Motif finding is very important in deciphering the biological meaning of DNA sequence data. It has many applications, such as restriction site detection and transcription binding site discovery. Biological approaches to this problem are tedious and time-consuming. The accumulation of large amounts of genomic sequence data and gene expression micro-array data have enabled researchers to solve this problem computationally.

Sequence motifs are often more complicated than substrings. They often consist of several substrings with varied length. For example, the Gal4p binding site is of the form $CGGN^{[1-11]}CCG$ where $N^{[1-11]}$ denotes a substring consisting of one to 11 arbitrary characters. Another common way to represent motifs is using a position weighted matrix (PWM). PWM has one row for each symbol of the alphabet, and one column for each position in the pattern. PWM score is the sum of position-specific scores for each symbol in the motif. For motif [AT]AN[AT][TGC], the PWM looks like figure 4.1.

Sometimes, a motif can be expressed as a regular expression ($CGG[ACGT]CCG$).

	1	2	3	4	5
A	0.4	1.0	0.3	0.5	0.0
T	0.3	0.0	0.2	0.5	0.3
G	0.2	0.0	0.3	0.0	0.3
C	0.1	0.0	0.2	0.0	0.4

Figure 4.1: Position Weight Matrix

Then the matches can be found using grep-like methods resulting in algorithms with $O(mn)$ running time if the regular expression has n symbols and the query string size is of size m.

Alternatively, motif matching can be reformulated as inexact string matching. One simple method of inexact string matching is by exact matching with wildcards: Given a motif $p = ab**c**ab**$. p_i is the set of maximal substrings of p that do not contain wildcards, and l is the array of starting positions in p of each of these substrings. (for our example $p_i = ab, c, ab$ and $l = 1, 5, 8$). A suffix tree approach can be applied to find all substrings in p_i in the search string t. We can then check if these matches are at the correct distance. The running time is $O(|p_i||t|)$[35].

Often, the motif information is not given knowledge. One of the biggest challenges in the field is to identify motifs among a set of related biological sequences.

4.1 Identification of motifs

Definition 4.1.1. *Let $s = s_1...s_m$ and $t = t_1...t_m$ be two character strings of the same length. The Hamming distance $d_H(s,t)$ of s and t is defined as the number of positions where $s_i \neq t_i$.*

Finding identical or similar substrings in a set of input sequences can be formulated as the following optimization problem:

Given a set of n strings $s_1, ..., s_n$ and a natural number I, find a $(n+1)$ tuple $(t, t_1, ...t_n)$, where t_i is a substring of s_i, and the string t is the motif to minimize the following cost function:

$$cost(t, t_1, ..., t_n) = \sum_{i=1}^{n} d_H(t, t_i) \tag{4.1.1}$$

It has been proven by Li that this optimization problem is NP-hard [55]. In practice, instead of simply finding all possible motifs, most algorithms try to find the motif that occurs more frequently than random. The reasoning behind this modification is as following:

1. Motifs tend to be short and in theory we would expect to match a motif with length 5 ($4^5 = 1024$) every 1000 bases.

2. Given that the human genome's size is 3 billion ($3 * 10^9$) and A, C,G,T distributes evenly (this is not the case in reality), any size 5 motif would have about $3 * 10^6$ matches on average.

3. Thus, it make sense to report size 5 motif A that matches 10^7 times but not size 5 motif B that matches $2.6 * 10^6$ times.

4.1.1 Exhausive Enumeration

One idea is to enumerate all possible patterns of a given size through the sequences and report the top ones based on certain criteria. Hertz [37] proposed to count all the possible patterns with size k through the sequences ($totalnumber = N$) and report the top ones. This algorithm guarantees the global optimum but the running time is $O(N4^k)$. Most of the 4^k possibilities are not relevant for any particular input sequence set since they do not occur

in any sequence at all. If it were optimized to only count all those words W that occur in at least one sequence [38], then the running time is improved to $O(N)$. However, if the true motif is weak, and does not occur exactly in any of the sequences, then this algorithm will never find it.

Tompa and Sinha [79] developed algorithm YMF that takes into account not only the absolute occurrence count but also the background distribution. The algorithm counts all the occurrences of substrings of size k as motif s allowing a small, fixed number c, of substitutions. Then all the motifs are scored base on a z-score that represents how unlikely it is to have N_s occurrences if the sequences were drawn at random according to the background distribution.

$$Z_s = \frac{N_s - Np_s}{\sqrt{Np_s(1-p_s)}} \tag{4.1.2}$$

Where p_s is the expected probability of motif s occuring at least once in one random sequence and M_s is the number of standard deviations by which the observed value N_s exceeds its expectation.

One thing that needs to be pointed out is that even though, the search space in an exhaustive enumeration approach grows exponentially with the length of the pattern (a relatively small number), the running time is linear with respect to the size of the input sequences. Thus the algorithm scales very well to larger gene families and longer promoter regions.

4.1.2 EM Algorithm

The idea of this approach is to try to partition the given sequences into two sub groups: pattern vs. background [61, 14]. Given a set S of genomic sequences. From this set S,

all k-long words x_1, x_2,..., x_n can be extracted. Each of these words can be seen as either motif or background. The objective of the algorithm is to maximize the log-likelihood of the probability that a certain classification (of pattern vs. background) is generated by the model.

Algorithm 2 EM algorithm

```
set initial values for p randomly
while p < ϵ: do
   compute Z from p (E-Step)
   compute p from Z (M-step)
end while
return p, Z
```

Given

- Z_{ij}: the probability that the motif in sequence i starts at position j.
- $p_{i,j}$: the motif's probability of having character i in position j.(from PWM). $p_{i,0}$ denotes the background probabilities.

The E-step:using p to compute Z

$$P(X_i|Z_{ij} = 1, p) = P1 * P2 * P3$$

$$P1 = \textit{probability\_of\_the\_characters\_before\_the\_motif}$$

$$P2 = \textit{probability\_of\_characters\_is\_part\_of\_the\_motif}$$

$$P3 = \textit{probability\_of\_the\_characters\_after\_the\_motif}$$

Following Bayes's rule on $P(Z_{ij} = 1|X_i, p)$, we can get Z_{ij}

The M-step: update p with the newly computed Z (the start position of each motif)

Since EM is a local optimization technique, there is no guarantee of finding the global maxima. In practice, EM algorithms are normally run from different start points to increase the chance of finding a globally optimal answer.

Chapter 5

Contribution

5.1 Contribution

In this study, I explore a number of analyses of DNA motifs in the Human Genome. Part II of this thesis, describes a new data representation for tandem repeats and a novel alignment-free clustering and classification method for tandem repeats that is made possible by the data representation. Analysis of clusters for tandem repeats in the human genome shows that the new method yields a good classification result in which similarity among repeats within a class is readily apparent. Furthermore, the classification result also shows potential to refine the original tandem repeats search algorithm.

Part III of this thesis, describes the construction of a collection of true transcription factor start sites through multiple data resources. and then presents the statistical analysis of combination of core promoter elements in multiple publicly available datasets. Through the combination of known core promoter elements motifs and a motif matching method that uses relaxation, high sensitivity was achieved.

Part II

Clustering and Classification of Tandem Repeats

Chapter 6

Clustering and Classification of Tandem Repeats

A tandem repeat in DNA is two or more contiguous approximate copies of a pattern of nucleotides. Tandem repeats, also called satellite DNA, are widespread in the human genome. The number of repeats varies between individuals but is often stable for each person and the same number of repeats get passed on from generation to generation. Tandem repeats can therefore fulfill the role of genetic markers and be used for a wide array of tasks including DNA fingerprinting [40], mapping genes, comparative genomics and evolution studies. When the repeats become unstable and undergo expansion (an increase in the number of repeats), clinical symptoms may occur once the copy number exceeds a certain limit (*figure 6.1*).

Trinucleotide repeat expansion diseases, caused by long and highly polymorphic tandem repeats of period size 3 [60], include over 30 hereditary disorders in humans, such as fragile X syndrome, myotonic dystrophy, Huntington's disease, various spinocerebellar ataxias, Friedreich's ataxia, and others [32]. In recent findings, microsatellites, i.e. tandem

Disease	Mutation/ repeat unit	Gene name (protein product)	Putative function	Normal repeat length	Pathogenic repeat length
Diseases that are caused by loss of protein function					
FRDA	$(GAA)_n$	*FRDA* (frataxin)	Mitochondrial iron metabolism	6–32	200–1,700
FRAXA	$(CGG)_n$	*FMR1* (FMRP)	Translational regulation	6–60	>200 (full mutation)
FRAXE	$(CCG)_n$	*FMR2* (FMR2)	Transcription?	4–39	200–900
Diseases that are caused by altered protein function					
SCA1	$(CAG)_n$	*SCA1* (ataxin 1)	Transcription	6–39	40–82
SCA2	$(CAG)_n$	*SCA2* (ataxin 2)	RNA metabolism	15–24	32–200
SCA3 (MJD)	$(CAG)_n$	*SCA3* (ataxin 3)	De-ubiquitylating activity	13–36	61–84
SCA6	$(CAG)_n$	*CACNA1A* ($CACNA1_A$)	P/Q-type α1A calcium channel subunit	4–20	20–29
SCA7	$(CAG)_n$	*SCA7* (ataxin 7)	Transcription	4–35	37–306
SCA17	$(CAG)_n$	*SCA17* (TBP)	Transcription	25–42	47–63
DRPLA	$(CAG)_n$	*DRPLA* (atrophin 1)	Transcription	7–34	49–88
SBMA	$(CAG)_n$	*AR* (androgen receptor)	Steroid-hormone receptor	9–36	38–62
HD	$(CAG)_n$	*HD* (huntingtin)	Signalling, transport, transcription	11–34	40–121
Diseases that are caused by altered RNA function					
DM1	$(CTG)_n$	*DMPK* (DMPK)	RNA-mediated	5–37	50–1,000
DM2	$(CCTG)_n$	*ZNF9* (ZNF9)	RNA-mediated	10–26	75–11,000
FXTAS	$(CGG)_n$	*FMR1* (FMRP)	RNA-mediated	6–60	60–200 (premutation)

Figure 6.1: Trinucleotide repeat expansion disease [32]

repeats with period sizes 1-6, have been shown to distinguish species [29] and moreover, to play an important role in cancer biology [30] : 18 high-similarity A/T rich repetitive motifs were found in the germlines and tumors of sporadic breast cancer and colon cancer tumor patients.

6.1 Related Work

Several software tools are available for finding tandem repeats in a sequence, some of which have been used to construct databases of tandem repeats. TRF [45] is the basis of TRDB [33]. TRed [82] is the software used in the TRedD database [81]. Other software tools include mreps [51] and ATRHunter [89]. A newly developed tandem repeat meta-search engine, TReads [64], allows a user to run several of the above software tools on a given sequence with similar parameters.

The multiplicity of tandem repeat finding software stems from the fact that tandem repeats in biological sequences are approximate repeats and that there are many different ways of modeling fuzziness in a repeat. Therefore, each of these software tools is based upon certain assumptions, even though most of them are somewhat flexible in that it is possible to modify parameters and affect the set of reported repeats. The approaches taken by these tools can be divided into two general categories:

- The first is a consensus-type approach, based upon the hypothesis that there exists some string called a consensus, which is similar to all copies in the repeat but is not necessarily an exact match to any actual copy. This approach yields a multiple alignment of the copies in the tandem repeat with the number of columns equal to the length of the consensus. Benson et al. in TRF [15] follows the consensus approach.

- TRedD [82, 81] uses another approach, based upon evolutive tandem repeats [34]. The assumption is that each copy is derived from a neighboring copy, possibly with mutations. Thus, each copy in the repeat is similar to its predecessor and successor copy, but there is not necessarily a consensus over all copies.

Both mreps [51] and ATRHunter [89] allow either the consensus or evolutive approach, and this is accomplished by adjusting particular parameter settings and the running mode.

6.2 Significance of Our Work

While each tool offers its unique insight into the repetitive sequences in a genome, very little effort has been put into the annotation and usability of the findings. The nature of tandem repeats, including their abundance, the presence of mutations, and rotational equivalence (TTATTATTA could be reported as TTA, TAT or ATT) makes this a difficult task. For example, in Chromosome 1 of Homo Sapiens, TRF locates 72,530 repeats, and TRedD locates 91,814 repeats. It is obvious that big text tables are not user friendly with respect to the presentation of this kind of data.

On the other hand, as shown in [29, 30], it is critical to have the ability to study the global content of tandem repeats across an entire genome, and to do so experimentally would require customized arrays that are very labor intensive and expensive. Our goal is to automate the classification of tandem repeats in a manner that facilitates the study of tandem repeats across an entire genome.

We address this problem by developing a new technique for organizing a dataset of tandem repeats that is independent of the original searching algorithms. We first come up with a novel representation of the tandem repeats that is independent of the definition of

the original tandem repeat-finding algorithm. We then propose a hybrid clustering schema on the reexpressed data to group the tandem repeats. Our prelimary results on chromosome 1 of the human genome show that the clustering method we propose yields well-defined clusters for which there is a defined similarity among the repeats in each cluster. We believe that through our method biologists will be able to visualize these repeats, make better use of the existing tools, and better utilize repeats for discovery and the advancement of biological science.

Chapter 7

Approach

Clustering and classification are commonly used techniques that group data into meaningful groups. They are used in many fields, including machine learning, pattern recognition, image analysis, information retrieval and bioinformatics.

Clustering is a widely used data mining technique [90, 17] in which data objects are grouped into sets, or clusters. Clustering has been applied to many bioinformatics applications [92, 87]. It is an unsupervised learning method, in the sense that it is intended for data with unknown categorizations. It is a natural fit for exploring the underlying structure of data sets such as tandem repeats, in addition to finding similar data (clustering) within the data set. Data within each cluster may be studied independently, and visualization of tandem repeats can then be dealt with easily.

Clustering itself is not a trivial problem and the end results depends on a series of choices that the user has to make:

- Deciding which features will be used to characterize the data
- Selecting a proper distance metric

- Choosing an algorithm
- Evaluating the results.

Indeed, the answers to those questions depend on the individual data set and intended use of the results. Therefore, cluster analysis is not an automatic task, but an iterative process of knowledge discovery or an interactive multi-objective optimization that involves trial and failure.

7.1 Feature Selection

Often the most important design decision when using standard clustering algorithms is in the data description. Features that can distinguish between different groups of data and those features that correctly identify important properties of the data are, of course, the best ones to be used. For tandem repeats, the choice of features used to express the data is especially important, as these are DNA sequences that already have very particular properties.

In my thesis work, I propose to summarize the sequence of a tandem repeat using the *n*-grams model. An *n*-gram [77] is a contiguous sequence of items of length *n*. They have been used in text classification and natural language processing in order to incorporate contextual information into the representation of text as feature/value pairs. For text documents, the items are usually words. In DNA sequences, however, each item is simply a letter from the set {A, C, G, T}, representing the different nucleotides. *n*-grams have also been applied to biological sequences in recent studies [87, 56].

Using the *n*-grams approach, with $n = 3$, each tandem repeat is re-expressed as a feature vector, $V = (x_1, x_2, ..., x_{64})$, where each x_i represents the normalized count of a

different trinucleotide in this particular sequence. Trinucleotides, or 3-grams, have been chosen to represent the DNA strings, because repeated trinucleotides are of special interest to biologists for many reasons [52]. Our experiments have also shown that 3-grams are an effective way of representing tandem repeats to facilitate comparisons between different repeats. Given the four letter alphabet $\{\text{A, G, C, T}\}$, there are 4^3 possible different 3-grams. Details on the relevance of the n-gram model to tandem repeats are provided in Section 8.1.

7.2 Distance Metrics

A second major decision for an appropriate clustering method is to determine which distance metric will be used to measure the proximity of data points.

Sequence metrics are often used to measure the similarity between DNA sequences. However, as pointed out in [16], standard sequence analysis techniques such as Smith-Waterman cannot be used for comparing tandem repeats. Thus, Benson defines a profile representation of a tandem repeat, which is a sequence whose length equals the number of columns in the multiple alignment and the elements are the character compositions within the columns. Building on this approach, Rao et al. [70] consider different possible distance functions for profiles, and cluster repeats according to the one that is shown to perform the best.

Our approach does not use the consensus pattern or profile, since for algorithms that do not define repeats according to a consensus pattern [82, 81, 34], profile representations do not exist. Furthermore, even if there is a defined pattern, a single repeat can often be broken down into different periods due to small errors (table 7.1). If such repeats (i.e. identical in sequence but different in period) were found at distant loci, they might not cluster together

according to a clustering scheme that compares profile representations. And yet, when examined closely, one can see that these repeats belong to the same family.

Start	*End*	*Period*
110832	110998	11
110832	110998	9
110832	110998	20
110832	110998	26

Table 7.1: OVERLAPPING REPEATS

In exact repeat finding, primitive repeats are defined to be the repeat with the smallest period. Repeats whose period is a multiple of another period spanning the same string, are not reported. For example, $(AT)^7$ can be viewed as period 2, 4, and 6, but is reported only as period 2. When searching for approximate repeats, this scheme cannot be used, since periods are often not multiples of one another. See Table 7.1 where 20 is not a multiple of 9 or 11, but close to being a multiple. A post-repeat finder should address this issue and cluster together repeats that are similar in sequence, but unrelated in period size. Counterintuitively, it is not possible to perform clustering strictly by checking repeats' overlap. First, we want repeats that are similar, but at distant loci, to fall into the same group. Second, a repeat may be a subrepeat of another repeat, and thus fully overlap, but be unrelated in terms of sequence (see table 7.2 where the TATATATA repeats should not be grouped with the larger repeat).

For vector space, there are in general two classes of distance (similarity) measurements; distance measures on the exact values vs. distance measures on the ranking of the values.

Euclidean distance and Cosine distance are the distance measures from the first class:

- Euclidean distance is the "classical" distance between two vectors. $d(q,p) = \sqrt{(q_1-p_1)^2+(q_2-p_2)^2+....+(q_n-p_n)^2}$

- Cosine distance measures the cosine of the angle between two vectors. $cos(p,q) = \frac{(q_1p_1)+(q_2p_2)+....(q_np_n)}{\sqrt{\sum_{i=1}^{n}(p_i)^2}\sqrt{\sum_{i=1}^{n}(q_i)^2}}$. The magnitude of the vectors won't affect the cosine distance measurement, thus providing a normalization effect, which we think may be beneficial to our data set.

Spearman similarity and Kendall similarity are the most well known ranking distance measurements:

- Spearman distance is the square of Euclidean distance between two rank vectors (i.e. using as input of the distance the rank of the values rather than the exact values).

- Kendall distance counts the number of pairwise disagreements between two ranking vectors.

Since the results of different distance functions are hard to predict, we propose to experiment with all of these classical distance metrics (Euclidean, Cosine, Spearman and Kendall) on our n-gram representation of tandem repeats. In Section 9.1 we give the results of experiments that compare the clustering results obtained using these different distance measures.

TATATATAGGGGGGGGGGGG**TATATAT**GGGGGGGGGGGT

Table 7.2: REPEAT EXAMPLE

7.3 Algorithms

Clustering is a very active research field and there are many different algorithms which can be listed as follows [91]:

- Hierarchical: Single linkage, complete linkage, group average linage, etc.
- Square Error-Based: k-means, partitioning around medics (PAM), etc.
- Kernel-Based: Kernal k-means, support vector clustering (SVC), etc.
- Large-Scale Data sets: CLARA, CURE, CLARANS, etc.
- Data visualization: PCA, ICA , etc.

Among these, Hiearachical Clustering (HC) and k-means-like clustering are the most studied groups of methods. The advantage of HC is that its result comes out structured as a binary tree which provides a very informative description and visualization of the data structure. The disadvantage of HC is that it suffers from a quadratic computational complexity in both running time and memory usage.

On the other hand, k-means-like clustering algorithms have running time $O(Nkd)$ and memory usage of $O(N + k)$. Given that N, the number of samples, is often much bigger than k, the number of groups, and d, the dimension of the space, this class of algorithms achieves almost linear performance. This is critical when clustering is applied to large scale data. K-means-like clustering algorithms also have many known shortcomings; there is no easy way to decide K, the mountain-climbing procedure of the algorithm sometimes reports only a local optimum, the algorithm is sensitive to outliers and noise, and it is only applicable to numeric data.

Given that our data set is of a massive scale (whole human genome) and in low dimension (4^3), we choose the k-means like method as our clustering algorithm. To overcome the known limitations, we decided to implement a hybrid clustering strategy by combining several modified algorithms at different stages of clustering to achieve optimal results:

1. The first step of our clustering approach is to apply the x-means algorithm on the n-gram features of the tandem repeats. The x-means algorithm is an extension of k-means[36], but unlike the traditional k-means algorithm, the user does not have to specify k, the number of clusters. x-means chooses k based on the maximization of the BIC (Bayesian Information Criterion) measure. Since we do not know the ideal number of clusters for the tandem repeat domain, this algorithm helps to understand the structure of the data. It is also a highly efficient algorithm and can easily handle very large data sets.

2. We then use the number of clusters output from x-means to guide the downstream k-means analysis. Specifically, using the statistical packages from R (http://www.r-project.org), we run the k-means like algorithm Clara (Clustering Large Applications) [45] through a stepping method: varying k around the number of clusters that x-means chose. Clara is an extension of PAM (Partitioning around Medoids) algorithm. The PAM algorithm is very similar to k-means, mostly because both try to break the dataset into groups and minimize error at the same time. PAM uses *medoids*, entities present in the dataset that represent the group in which they are inserted, while k-means works with *centroids*, artificially created entities that represent each cluster. In order to work on large data sets, Clara extends the PAM algorithm through sampling. Instead of finding medoids for the entire dataset, CLARA draws

a sample from the dataset and applies the PAM algorithm to generate an optimal set of medoids for the sample.

7.4 Evaluation

Given a dataset, a clustering algorithm can always converge to a final result even if the sub-structure does not exist. Moreover, different approaches usually lead to different clusters; and even for the same algorithm, parameter settings and ordering of the input data may effect the final results. Therefore, effective evaluation standards and criteria are important to provide the users with a degree of confidence for the clustering results derived from the used algorithms [91].

Both Akaike information criterion (AIC) [8, 9] and Bayesian information criterion (BIC) [76] are measures of the relative quality of a statistical model for a given set of data. They both deal with the trade-off between the complexity of the model and the goodness of fit of the model. $AIC = 2k - 2\ln(L)$ and $BIC = -2 * \ln(L) + k\ln(n)$, where k is the number of parameters and L is the likelihood of a statistical model for the given data n. In k-means like clustering algorithms, they are often used to evaluate which k is a better fit for the data.

Average Silhouette Width (ASW) [74] is the measurement of the ratio between inter-cluster distance and intra-cluster distance. $ASW = \sum_{i=1}^{n} \frac{b(i)-a(i)}{max(a(i),b(i))}/n$. A good clustering result should be the one with minimum intra-cluster distance and maximum inter-cluster distance, resulting in an ASW value closer to 1. In practice, $ASW > 0.7$ means excellent clustering and $ASW > 0.5$ means a reasonable structure has been found. In Section 9.1, we present our clustering results with ASW measurements.

Chapter 8

Method

8.1 Ngrams on Tandem Repeats

Repeats differ from standard DNA sequences, since some specific n-grams that are part of the repeated sequence will be much more common in the repeats than other n-grams. Furthermore, many of the possible n-grams will have zero-frequency, since they are not part of the repeat, yielding a sparse vector with only a small number of large values. Most importantly, the limitation of n-grams, in that they lose information on long range dependencies, is not nearly as pronounced when representing repeats. Since the repeated strings are in fact already representing the long range context, it is inherently modeled even when n is small, as those short n-grams are repeated throughout the length of the sequence. An n-gram model for tandem repeats therefore retains more information than n-grams for non-repetitive sequences.

We illustrate this by first considering unigrams. Suppose we are given a tandem repeat with $50\%A$ and $50\%T$. The sequence cannot be any random distribution of As and Ts but rather it must be alternating fixed length sequences of As and Ts, such as represented by

the regular expression $[A^xT^yA^yT^x]^+$. As such, simple unigram frequencies can tell something about the order of bases in a tandem repeat. When considering 3-grams, much more contextual information is provided since we consider all overlapping sequences of length 3. Considering the same example, the count of the trinucleotides AAA, TTT, TAA, AAT, and ATA give information about the values of the exponents and the number of periods. As an additional example, consider two repeats: $(AGTCCT)^{20}$ and $[(AGT)^{10}(CCT)^{10}]^2$, both of length 120. The unigram and bigram frequencies for the two repeats are identical. Yet tri-grams incorporate contextual information. TABLE 8.1 shows all trigrams that have non-zero values. The differences in the CTA, CTC, GTA, GTC counts represent the borders between the periods, which is the key difference between the two repeats.

Period	*Repeats*	*AGT*	*CCT*	*CTA*	*CTC*	*GTA*	*GTC*	*TAG*	*TCC*
6	$(AGTCCT)^{20}$	20	20	19	0	0	20	19	20
60	$[(AGT)^{10}(CCT)^{10}]^2$	20	20	1	18	18	2	19	20

Table 8.1: REPEATS

In the next example, shown in TABLE 8.2, we examine a repeat, with period size 11 that contains all four trinucleotides. However, at closer examination, it seems plausible that the Cs and the Gs are mutations. This is summarized quite effectively in the 3-gram sequence since each trinucleotide that contains a C or G has the value of 1. If we ignore the low values, assuming they are errors, the repeat contains the trinucleotides: ATA, ATT, TAT, and TTA. Although period size 11 is unrelated to 3, the counts of the trinucleotides tell us about the sequence in each period. Specifically, there are 3 copies of the period, and 3 ATTs since there is one ATT in each copy. Furthermore, there are 9 TATs which tells that each copy is similar to a TATA sequence. This example showed that a 3-gram vector

contains a great quantity of information, including information regarding periods unrelated to size three.

Start		*End*
110887	ATATCTATTAC	110897
110898	ATATATATTAT	110908
110909	ATATGTATTAT	110919

Table 8.2: REPEAT WITH ALL FOUR NUCLEOTIDES

8.2 Data Set and Data Cleaning

In this work, we first applied clustering techniques to the full tandem repeat set from Homo Sapiens chromosome 1, as obtained from the TRedD program [81]. This set contains 91,814 tandem repeats. We removed all repeats that consist of a single nucleotide (i.e. polyA,G,C,T tracts) since these are trivial to cluster into 4 clusters, each corresponding to a specific nucleotide. TABLE 8.3 lists the mean, median, maximum and minimum information on the length, period size, and number of copies for the full remaining set of data of size 83,591.

No other preprocessing on this large set was done prior to clustering. Previous work in DNA or tandem repeat clustering [70, 87] used preprocessing as a way of pruning the dataset to remove overlapping repeats or as a way of choosing those sequences that were thought to best represent distinct clusters. Our focus has been different. We make no a priori assumptions about which tandem repeats are most important, which should be removed before clustering, or which should be grouped together. We used the results of the clustering algorithm to develop a hierarchical classification technique which was applied

to the entire set of tandem repeats in the human genome. Table 8.4 lists the mean, median, maximum and minimum information on the length, period size, and number of copies for the full set of tandem repeats in the human genome.

Finally, we validated our methods using tandem repeats data from the human genome from TRDB [33]. We purposely chose our training data set and validation data set belonging to the different categories that were described in Section 6.1 to make sure that our method works universally on the two groups of algorithms. We show a case study on a set of repeats with AT-rich regions that were shown to be common in cancer patients.

	Min	*Median*	*Mean*	*Max*
Length	20.00	42.00	66.82	3576.00
Period	1.00	4.40	12.68	459.50
Copies	2.00	8.00	12.66	443.00

Table 8.3: Chromosome 1 Tandem Repeat Statistics

	Min	*Median*	*Mean*	*Max*
Length	20.00	41.00	69.82	18,617.00
Period	1.00	4.20	12.84	499.00
Copies	2.00	8.80	13.40	443.00

Table 8.4: Whole Genome Tandem Repeat Statistics

We did experimentation on the full data set (**All64 Norm**), to demonstrate the significant value concept. We also show results on cleaned data with only the top 3 values in each vector (**Top3 Norm**), setting the rest to zeros. As an additional set we used only the boolean value for the top-three trinucleotides in each tandem repeat (**Top3 Bool**), instead of a normalized count. In this case, each vector V contains only three non-zero entries, and

each of those non-zero entries equals 1. This simple representation of repeats can be expressed in English as, which are the top three trinucleotides in the repeat? A very natural clustering or a classification of the tandem repeats into similar sets can be computed easily and compared to the two other representations. Keeping the top three trinucleotides also allows for the full expression, in a cyclical sense, of the well-known disease-related repeat triplets.

8.2.1 Significant Values

In order to classify tandem repeats it is desirable to ignore small values in the trinucleotide vector representation of a tandem repeat. This would in essence allow for a small amount of mutation. Clearly, a trinucleotide that occurs in a tandem repeat exactly once, should not affect the summarization of the repeat sequence. Similarly, any value that is very small relative to the rest, should be considered insignificant. Therefore, our goal was to determine a reasonable definition of what constitutes a significant value in each vector. In order to obtain a baseline for comparison, we first plotted the distribution of the number of non-zero values in each vector. We then considered several possible thresholds, based upon the ideas in the following discussion. Consider the following repeat, where the copy number is 100: ACCT AGCT ACCT ACGCT ACCT ACCT. In this case, the fact that GCT occurs twice is insignificant, and is not enough to ignore values less than or equal to one. This led us to the idea of using a percentage of the copy-number, i.e. the number of adjacent repeated units within a tandem repeat. We consider a value significant if it constitutes at least 75% of the copy-number, that is the trinucleotide appears in 75% of the copies of the repeated pattern. It is intuitive that the trinucleotide should occur in at least $3/4$ of the copies to be considered to be a significant part of the repeat. However, this definition is problematic, since although

it works well for short repeats with several copies, this definition is problematic when the copy-number is small, i.e. the repeats are longer with fewer copies. For example, for a repeat with period 100 repeated 2 times, all values would be considered significant according to this definition. As an alternative, we considered using the length of the repeat as a factor in defining significant values in a repeat. A value is significant relative to the length of the repeat, so it makes sense to set a threshold for significant values as a percentage of the length. For example, given the repeat AAAAAAAAAAAAAAAAAGAAG, with a length of 21, the threshold of 70% of length results in a good classification of the repeat. Values in the vector are AAA with 15, AAG with 2 and AGA with 1. 70% length = 14.7, so only AAA is significant, which is intuitive. However, according to this definition many vectors will have zero significant values, a situation that may be best avoided. Whenever all the values in the vectors are close together and below the threshold for the percent of the length, all of its values will be considered insignificant. This can be illustrated by the repeat of length 21: TAAATAAATAAATAAATAAAT The values in the vector are AAA with 5, AAT with 5, ATA with 4 and TAA with 5. All of these values are below the threshold of 70% length and even 25% length and the repeat will have zero significant values. Clearly, the definition of significant values cannot depend solely on the length of the repeat. This led us to identify another feature of the repeat that is crucial: the maximum value in the trinucleotide vector. The reasoning behind this is that a value is significant relative to the maximum value; if a value is a small percentage of the max value it is insignificant. This approach helps avoid the problem of using a percentage of the length, since it relates the significance of a value to the max, which is a value in the vector, and therefore does not result in zero significant values. After experimenting with different thresholds using percentages of the max value, we concluded that the best option is to use a combination of

the copy-number and the max value in the repeat in defining significant values. Thus, we settled on the formula, value is significant if and only if:

$$value > \frac{MAX}{3} \| value > \frac{3}{4} copynumber$$

The distribution of this formula is very similar to the non-zero baseline. As shown in Table 8.5, according to this definition of significant, there are no repeats that have zero significant values. Moreover, in all of the chromosomes (except for chromY) more than half of the repeats have 1-4 significant values in the vector. Thus, it is justified to focus on vectors with few significant values, since this includes a majority of the repeats in the genome. In the next section we use a hierarchical approach to classify repeats according to their topX significant values, where X ranges from 1-10.

8.3 Clustering Strategy

We applied a hybrid clustering strategy by first running the x-means algorithm [63], then Clara (Clustering Large Applications)[45] multiple times, varying k around the number of clusters that x-means chose. Clara is an extension of the PAM (Partitioning around Medoids) algorithm, a more robust version of the k-means algorithm, and deals with large data sets through sampling. Experimenting with different values for k allows us to compare the quality of the clusters using the ASW [-1,1]: a value closer to 1 indicates a good clustering.

chrom	*numRepeats*	$sigval = 1$	$sigval = 2$	$sigval = 3$	$sigval = 4$	$sigval > 4$
chrom1	91814	29%	13%	6%	20%	32%
chrom2	92525	25%	12%	5%	18%	40%
chrom3	69829	27%	13%	6%	20%	34%
chrom4	69485	28%	12%	6%	20%	34%
chrom5	65195	24%	15%	6%	19%	36%
chrom6	62481	28%	13%	6%	19%	34%
chrom7	66935	31%	13%	6%	19%	31%
chrom8	55218	27%	10%	5%	18%	40%
chrom9	49231	33%	10%	6%	20%	31%
chrom10	59749	25%	14%	6%	20%	36%
chrom11	49759	32%	8%	5%	20%	35%
chrom12	55029	26%	14%	6%	19%	35%
chrom13	35496	26%	11%	5%	19%	38%
chrom14	35187	21%	11%	5%	16%	47%
chrom15	33076	28%	8%	5%	18%	41%
chrom16	44230	29%	14%	6%	20%	31%
chrom17	40669	24%	14%	6%	19%	37%
chrom18	28732	26%	14%	6%	20%	34%
chrom19	38593	27%	14%	6%	20%	33%
chrom20	27558	27%	12%	6%	19%	36%
chrom21	17256	26%	13%	6%	20%	35%
chrom22	20880	28%	13%	6%	20%	34%
chromX	60144	23%	15%	6%	21%	34%
chromY	13092	15%	10%	4%	15%	57%

Table 8.5: Percentage of Repeats with 1-4 and > 4 Significant Values

8.4 Hierarchical Classification of Whole Genome Tandem Repeats

We implemented a top down, tree-like classification schema to enable the end user explore the tandem repeats with few significant values. Since over 80% of the tandem repeats in the human genome have $<= 10$ significant values, we ran the classification schema for the first 10 levels. Given n tandem repeats, let s_i be the number of significant values of tandem

repeat i. We classify the set of tandem repeats for levels $1 <= j <= 10$. At level j, we classify all the repeats such that $s_i >= j$ based upon their top j tri-nucleotides.

For example, consider the five repeats in Table 8.6 (n=5).

ID	*loci*	*sigval*	*label*	*sequence*
Repeat1	chr6:134589525-134589565	$s_1 = 1$	AAA	AAAAAAAAAAATAAAAATAAAAATAAATAAATAAAAAATAA
Repeat2	chr10:20436002 -20436047	$s_2 = 2$	TATA.ATA	TATATATATAGTATATATATACACTATATATATACTATAGTATATA
Repeat3	chr11:122998574-122998600	$s_3 = 2$	ATA.TAT	ATATATATATATAAATATATATATATA
Repeat4	chr11:128708186-128708215	$s_4 = 2$	ATA.TAT	AGAGATATATATAAATATATATATATATAT
Repeat5	chr2:13886142-13886161	$s_5 = 3$	TAT.ATA.TTT	TATATATATATATTTTTT

Table 8.6: Example Repeats

The Hierarchical Classification at the different levels will be as follows:

For level 1 , $j = 1$ we classified all tandem repeats based upon their top 1 tri-nucleotide label. We would have four classes: Repeat1 in Class AAA; Repeat 2 and Repeat 5 in Class TAT; Repeat 3 and Repeat 4 in Class ATA.

For level 2 , $j = 2$ as Repeat 1 only has one significant value ($s_1 = 1$) it will be filtered out before this stage. Repeat 2, Repeat 3, Repeat 4 and Repeat 5 are all in Class ATA.TAT, as we only care about the top 2 significant values; order is not taken into account.

For level 3 , $j = 3$ only Repeat 5 will be classified and it will be in Class ATA.TAT.TTT

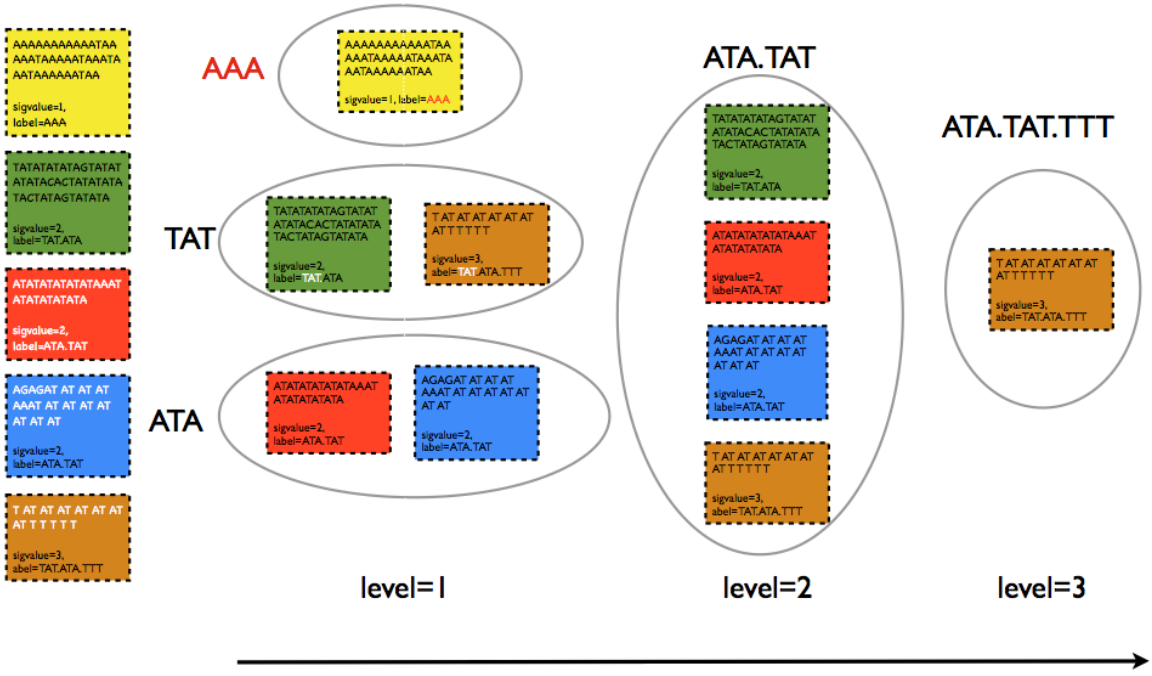

Figure 8.1: Hierarchical Classification

Chapter 9

Results

9.1 Clustering Results on Chromosome 1

We report the Average Silhouette Width (ASW) for each Clara clustering run on the tandem repeat dataset. N-gram clustering often uses geometric distances such as the Euclidean measure used in Clara. Table 9.1 presents results for different values of k, the number of clusters, that range from 50 to 1000. We report values for the three different representations of the dataset; normalized counts on all 64 trinucleotides (**All 64 Norm**), normalized counts of only the top three trinucleotides (**Top 3 Norm**), and top three trinucleotide n-grams as boolean features (**Top 3 Bool**). A very important point of this work is to demonstrate that the clustering methodology is independent of the repeat-finding algorithm that created the tandem repeat database. For validation, we used the TRDB database [33] which is a consensus based approach to finding repeats. We ran Clara, using the top-3 boolean representation over the same range of k, on the tandem repeat database of chromosome 1. We compare the ASW for both sets of data in Table 9.2. As can be seen from all the rows,

including the bold-faced rows, the ASW values for both datasets are similar, and both data sets have $ASW > .50$ on identical values of k.

k value	*All 64 Norm*	*Top3 Norm*	*Top 3 Bool*
100	0.32975	0.30125	0.48077
200	0.33817	0.26013	**0.64773**
500	0.35258	0.22494	**0.59143**
750	0.32418	0.20689	**0.61115**
1000	0.33509	0.20254	0.35196

Table 9.1: AVERAGE SILLOUETTE WIDTH

k value	*TRF Rpeats*	*TReD Repeats*
100	0.43570	0.48076
200	**0.66265**	**0.64772**
500	**0.63962**	**0.59143**
750	**0.60359**	**0.61115**
1000	0.36667	0.35196

Table 9.2: ASW FOR DIFFERENT TANDEM REPEAT DATASETS

Distance Measure	*Spearman*	*Cosine*	*Euclidean*	*Kendall*
ASW k= 200	**0.383022**	0.28177	0.23637	**0.389432**
ASW k= 500	**0.428520**	0.27557	0.21036	**0.428518**

Table 9.3: DISTANCE MEASURE COMPARISON

9.2 Distance Measures

The next step in our experimentation considered the full 64 feature dataset to explore whether different distance measures are more appropriate for clustering tandem repeats

via the 3-gram model. Specifically, our experiments chose multiple random sets of size 5000 from the full set of repeats, varying k from 30 to 500, and compared the results of four different distance functions. Table 9.3 shows that clustering using the Spearman and Kendall distances had a much higher ASW than other methods. We concluded that ranking correlation distances perform better on DNA repeats. Similar results have been reported [87] showing that using a rank correlation distance measure to compare n-grams of DNA sequences creates meaningful clusters.

The experiments on distance measures led us to combine the two ideas that best represents our clustering of tandem repeats which is to focus on the top-3 trinucleotides in each repeat in conjunction with a ranking distance. This can be viewed as a simple classification problem, with each tandem repeat labeled with its top 3 trinucleotides. This ranking distance incorporates the well-known triplet disease information in tandem repeats and in the next two sections we show that it results in well-defined clusters that enable further investigation and evaluation.

9.3 Top-3 Classification Results

We present properties of the clustering scheme based upon classification of tandem repeats by their top three 3-grams. Analysis of these clusters for chromosome 1 of the human genome shows that the clustering of tandem repeats according to 3-grams yields well-defined clusters in which there is a definite similarity among the repeats within each cluster. Although the method is alignment-free, the similarity within the clusters is based upon sequence similarity, and it is not related to period size or to genomic location. We do not measure ASW to validate the clusters since this scheme is basically an equivalence relation, thus $ASW = 1$. However, we highlight specific attributes of the clusters in chromosome

1 to demonstrate the efficacy of this method. In Section 9.4 we provide some interesting examples displaying uniformity within the clusters. Since there are 4^3 trinucleotides, there are $C(4^3, 3) = 41,664$ possible classes or clusters, but for our dataset the number of clusters is only $8,753$. Of these clusters, $5,254$ had only one repeat, that is, $78,337$ or 94% of the data is clustered in $3,499$ non-singleton clusters. Clusters that contain more than 5 elements included 86% of the repeats. We summarize the cluster sizes in the graph in Figure 9.1. The left portion of the graph shows that there are fewer large clusters, yet the graph on the right displays that the majority of repeats are included in large clusters.

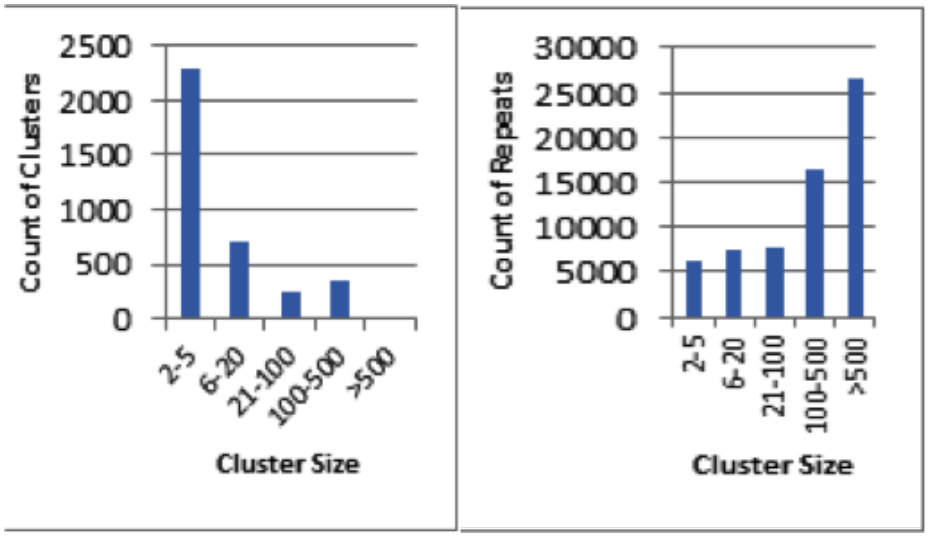

Figure 9.1: Cluster Size

The next point of investigation dealt with relating the repeats overlap on the genome to the clustering scheme. As mentioned previously, simply overlapping in the genome should not cause repeats to cluster together. We used BEDtools [69] a collection of utilities that allows one to address common genomics tasks to find pairs of repeats that overlap, with padding options of 10, 100, and 1000. The results for the different padding options were

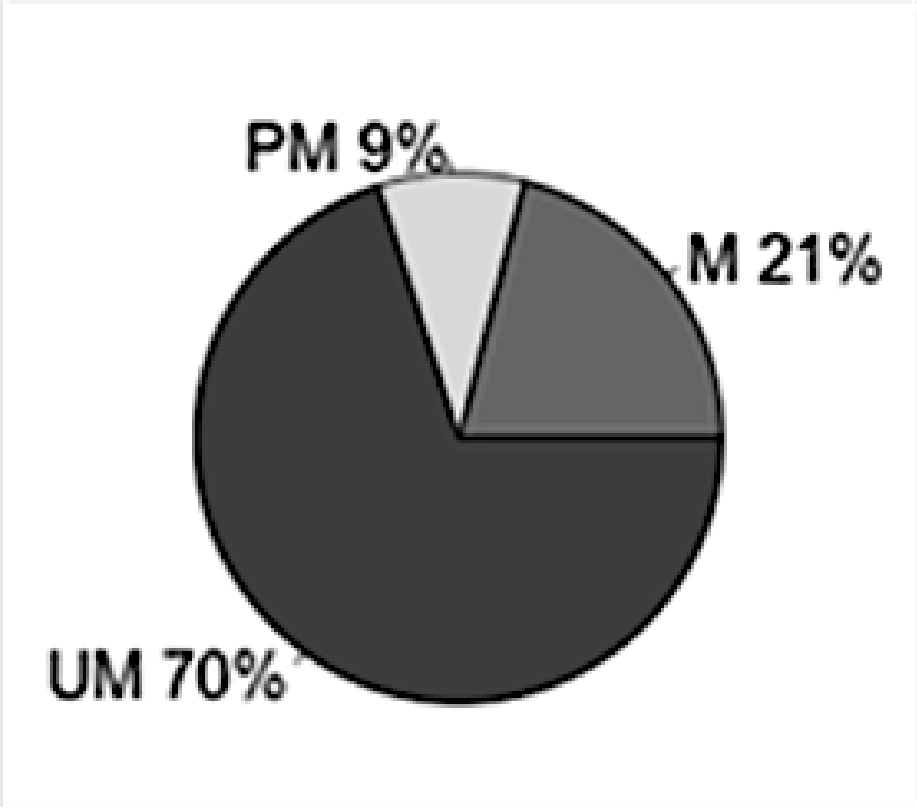

Figure 9.2: Overlapping Repeats

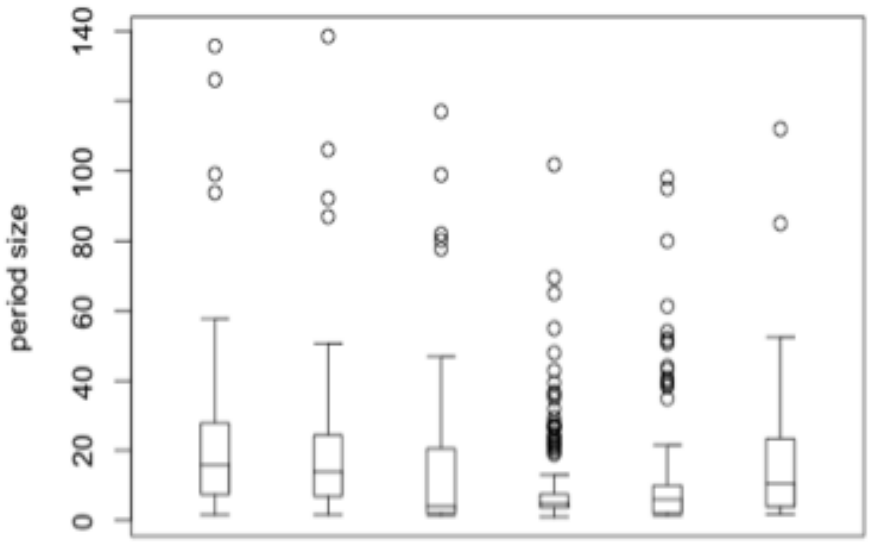

Figure 9.3: Period Distribution of Large Clusters

almost identical; hence we report the results for pairs of repeats that overlap within a flanking window of 100bp. The pie-chart in Figure 9.2 shows the percentage of a match(**M**), a partial match(**PM**) and an unmatch(**UM**) for pairs of close repeats. Match(**M**) means that the repeats have the identical top3 3-grams, **PM** means that they have the same top3 but in a different order, and **UM** means that the top3 3-grams of the repeats differ. Note that in our scheme, both **M** and **PM** cluster together. The graph shows that 70% of repeat pairs that overlap on the genome do not cluster together. We can conclude that our clustering scheme is not dependent on the loci of the repeats. As a final analysis of the clustering scheme, we studied the distribution of period size in the large clusters. We want to confirm that our clustering scheme does not cluster only identical period sizes together, but rather clusters similar sequences with possibly varying period sizes. We report the distribution for several of the large clusters in Figure 9.3. It is apparent that there is a spread of period sizes in the clusters.

9.4 Example Clusters

In addition to the statistical analysis performed on the clusters, we examined the actual repeat sequences in many of the clusters. In this section we report on the sequence similarity observed within the clusters. We began by studying the clusters of disease-related trinucleotides. In Table 9.4 we report the disease-related trinucleotides with the number of elements in each cluster, where the top 3 trinucleotides are the 3 rotations of each disease triplet, e.g. CGG, GGC, and GCG.

Disease	*Trinucleotide*	*Number of elements in TredD*	*Number of elements in TRDB*
FRAXA	CGG	70	66
FRAXE	GCC	79	48
FRDA	GAA	437	528
DM, SCA8	CTG	73	55
PolyQ Diseases	CAG	105	67

Table 9.4: DISEASE RELATED REPEATS

Length	*Period*	*Repeat sequence*
25	4	GAGCGAGCGGGCGGGCGGGCGGGCG
29	6	GGCGGAGGCGGAGGCGGAGGCGGAGGCGG
25	9	CGGCGGCAGCGGCGGCAGCGGCGGC
21	3	GCGGCGGCGGCGGCGGCGGCG

Table 9.5: EXAMPLE REPEATS IN CGG CLUSTER

To illustrate a specific example, we show several sample repeats in the CGG.GGC.GCG cluster in Table 9.5. This cluster should represent all CGG-rich regions. Since this set of three trinucleotides is essentially CGG repeated, most, although not all of the instances in this cluster have a period size that is a multiple of 3. Although one may conclude that many

of the repeats in this cluster should originally have been reported with period size 3, due to particular features of the alignment, mutations, and the repeat-finding program, this was not the case. As such, the clustering scheme refines the original repeat finding algorithm. We further point out that the first row has a period of 4 since there are 3 Gs between the Cs, yet it still has the CGG cyclic permutations as its top3 3-grams.

We next examined the repeats that were placed into large clusters. It turns out that many of the repeats are relatively simple sequences and are thus well represented by their top 3 trinucleotides. In Table 9.6. we show all clusters that have a size greater than 2000. These clusters are made up of a single nucleotide with a second nucleotide at regular intervals. For example, the third largest cluster, AAA.AAC.ACA, reflects the nucleotide composition of a sequence of As with regularly interspersed Cs, containing more As than Cs. We can summarize this cluster with the regular expression $(A^+C)^*$. For each repeat, the + can be replaced by a particular number, as can be seen in some sample repeats in Table 9.7. In a similar manner, most of the remaining large clusters are nicely summarized with a simple regular expression.

Top3 Trinucleotides	*Number of Elements*	*Regular Expression*
AAA.AAG.AGA	4801	$(A^+G?)^*$
CTT.TCT.TTT	4516	$(T^+C?)^*$
AAA.AAC.ACA	2880	$(A^+C?)^*$
AAA.AAT.ATA	2661	$(A^+T?)^*$
GTT.TGT.TTT	2656	$(T^+G?)^*$
ATT.TAT.TTT	2300	$(T^+A?)^*$

Table 9.6: REGULAR EXPRESSIONS FOR LARGE CLUSTERS

Finally, we examined several clusters with more complicated sequences such as those that contain 3 or 4 distinct nucleotides. These clusters also displayed interesting properties.

For example, consider the cluster GTG.TGT.AGA containing 92 repeats. Most repeats in this cluster have period size of 2, reflecting repeats with a small period size that changed into another repeat with the change of one trinucleotide. The regular expression that summarizes this cluster is $(GT)^{+}(GA)^{+}$. See Table 9.8 for an example of one such repeat in this cluster.

Length	*Period*	*Repeat sequence*
27	4	AAACAAACAAACAAACAAACAAACAAA
24	7	AAAAAACAAAAAACAAAAAACAAA
40	5	AAAACAAAACAAAACAAAACAAAACAAAACAAAACAAAAG

Table 9.7: Example Repeats for $(A^{+}C)^{*}$

	Start Pos	*End Pos*	*Length*	*Period*
Repeat Statistics	204775641	204775713	73	2
Repeat Sequence	GTGTGTGTGTGTGTGTGTCTGTGTGTGTGGTGTGTGTGTGTGTGTGTGTGTGAGAGAGGAGAGAGAGA			

Table 9.8: Tandem Repeat with Change

9.5 Whole Genome Hierarchical Classification Results

Since 65% of the tandem repeats in the human genome have fewer than 5 significant values in their vector representations, in Figure 9.4 we show the results for the first four levels of the hierarchical classification. The results show that the majority of the repeats have simple sequences that can be summarized nicely with only a few trinucleotide counts. A simple regular expression can represent each class of tandem repeat.

For level one classification, the top classes are TTT and AAA, which is due to polyA and polyT elements. We show, in Figures 9.5, 9.6, 9.7, 9.8, the size distribution of the

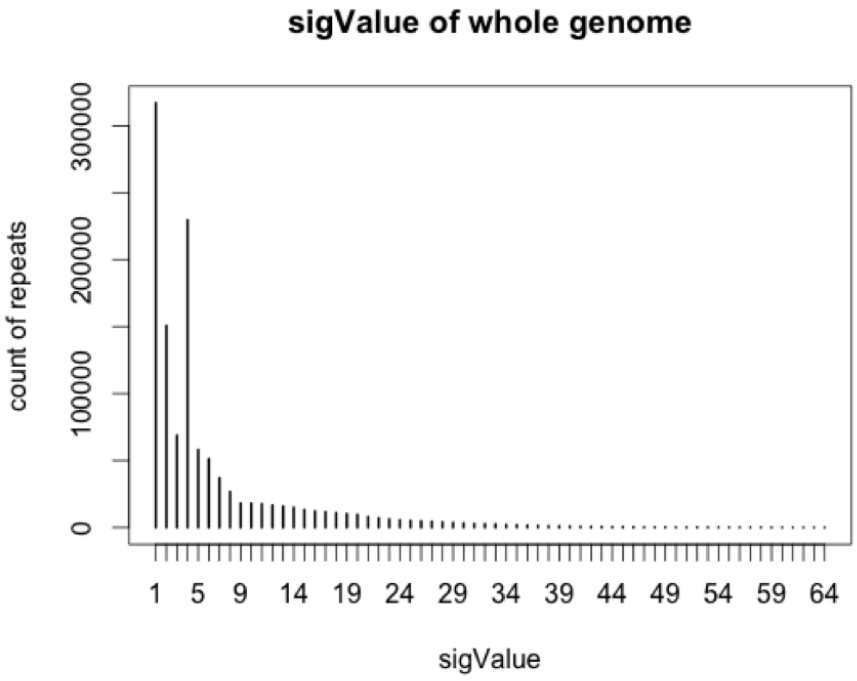

Figure 9.4: Distribution of the number of significant values

clusters for the top 4 levels of the hierarchical classification. Note that the majority of the repeats are classified in very few large clusters. This was an interesting and surprising result, as a priori we had no idea of what kind of sequences were common among the tandem repeat data set.

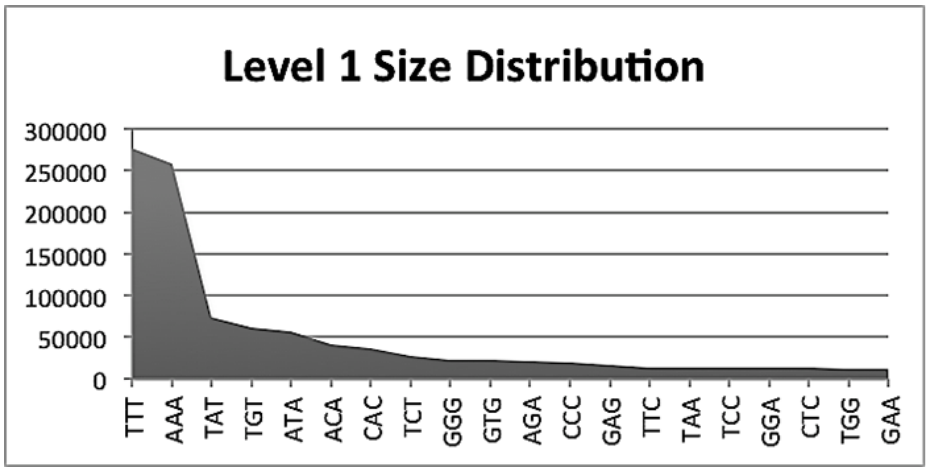

Figure 9.5: Size distribution of the clusters for the first level of the hierarchical classification

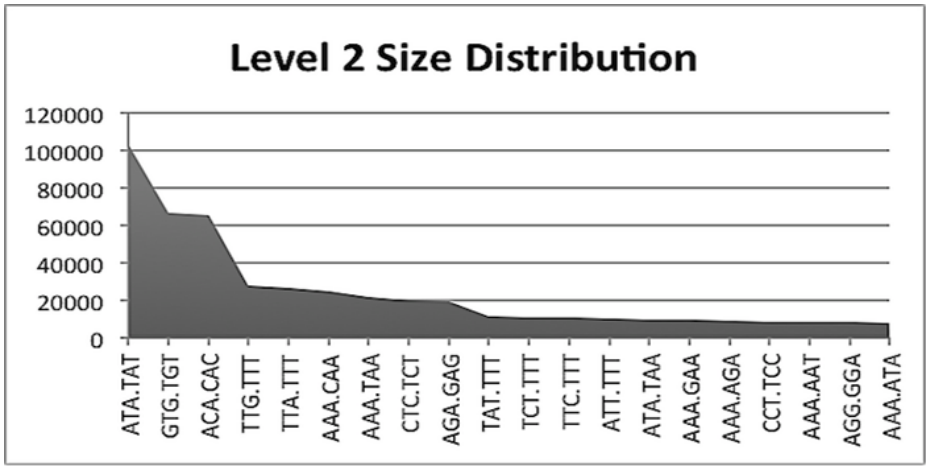

Figure 9.6: Size distribution of the clusters for the second level of the hierarchical classification

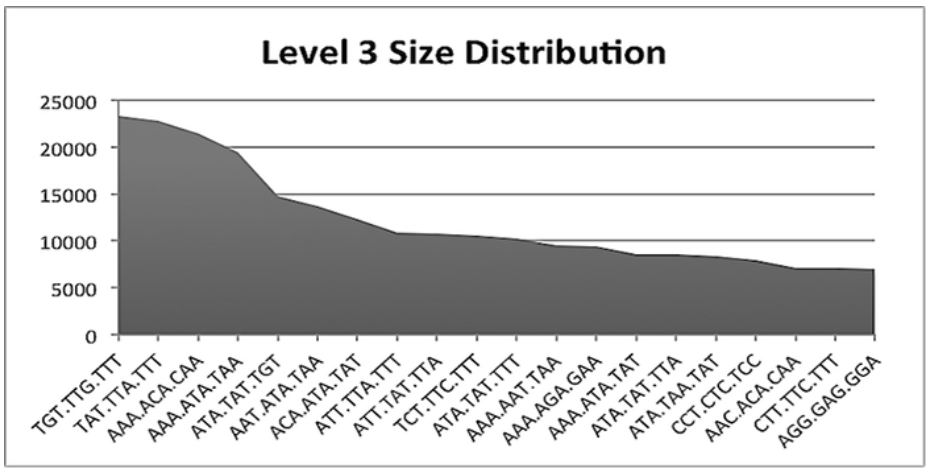

Figure 9.7: Size distribution of the clusters for the third level of the hierarchical classification

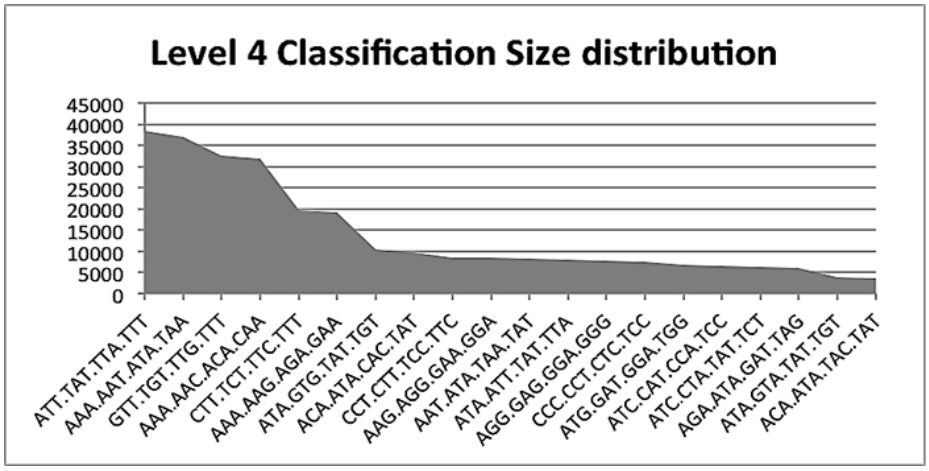

Figure 9.8: Size distribution of the clusters for the fourth level of the hierarchical classification

9.6 Case Study

We then examined more closely a set of tandem repeats (cancer repeats): $TATT$, TTA, $TATTT$, $AATTTT$, $AATT$, $TATATT$, $TATAT$ that were found to play an important role in cancer biology[30]. Several tandem repeat searching algorithms do not report a consensus pattern. For those that do, the reported patterns are largely influenced by the parameter setting. Thus the pattern of a tandem repeat is not a reliable searching point. Hence, given a set of biologically related repeats, it is not trivial to parse out all the tandem repeats that belong to this group. In fact, the original paper that discovered those tandem repeats located them through a complicated fuzzy matching perl script. We implemented our own fuzzy search script and located all the tandem repeats that have at least three copies of the cancer repeats from TRDBs [33] chr1 data. There are 3,034 of them, with 1,541 representative TRDB consensus patterns. Among the 1,541 different reported TRDB patterns, only 17 of them actually represent more than 10 cancer repeats (as shown in Table 9.9). These 17 patterns represent a total of 1,255 repeats, less than half of the 3,034. Thus, even if one goes through all the 17 patterns, only less than 50% of all cancer repeats can be recovered. On the other hand, looking at the cancer repeat distribution on level 2 of our classification, we can see that repeats are much better clustered; with a total 45 different classes. Details of this are in Table 9.10. It should be noted that the top 4 classes recover more than 80% of the searched repeats. Furthermore, Class TTA.TTT by itself represents almost 50% of the cancer repeats. Once looking at the Class TTA.TTT carefully, we can also see that of all 1,375 repeats that were reported, 88 are not cancer repeats. This means that our method not only has high sensitivity, but also very good specificity. From this case study of cancer relevant repeats, we have demonstrated that our method greatly increases

the usability of the current existing tandem repeat database and has the potential to facilitate the mapping between tandem repeats and their biological functions.

TRDB Pattern	*Count*
TTTA	430
TTAT	154
ATTT	149
TATT	117
TTTTA	98
TTTAT	50
ATTTT	49
AT	37
TATTT	35
TA	33
TTATT	30
TTTTTA	20
ATTTTT	11
TTTTAT	11
TTTTTTA	11
TTTATTTATTTATT	10

Table 9.9: Tandem repeat patterns from TRDB of cancer repeats

Class	*Cancer repeats count*	*Total count of class*	*Percentage of all cancer repeats*
TTA.TTT	1284	1375	42%
ATA.TAT	542	6627	18%
ATT.TTT	349	382	12%
TAT.TTT	334	472	11%
TAT.TTA	87	338	3%
ATT.TTA	80	202	3%

Table 9.10: Statistics for Cancer Repeats from Classification Result

Part III

Core Promoter Elements On High Throughput Data

Chapter 10

Background & Introduction

A human is made of cells with widely differing characteristics such as muscle, blood and neural cells. These characteristics are specified by genes, and yet each cell contains the same set of genes [68]. What distinguishes hands from feet is the expression of common genes at different times, in different places and in different combinations.

Transcription ($DNA \rightarrow RNA$) is the first step of gene expression that controls how much RNA is produced. During transcription, the information contained in the specific sequence of DNA is translated into a corresponding sequence of message RNA (mRNA). The mechanism of transcription in a eukaryotic [1] cell, particularly the human cell is much more complex and more tightly controlled compared to prokaryotes [2].

[1] eukaryote is a group of organisms whose cells contain a nucleus and other organelles enclosed within membranes.

[2] prokaryote is group of organisms lack of true nucleus and other membrane-bound cell compartments

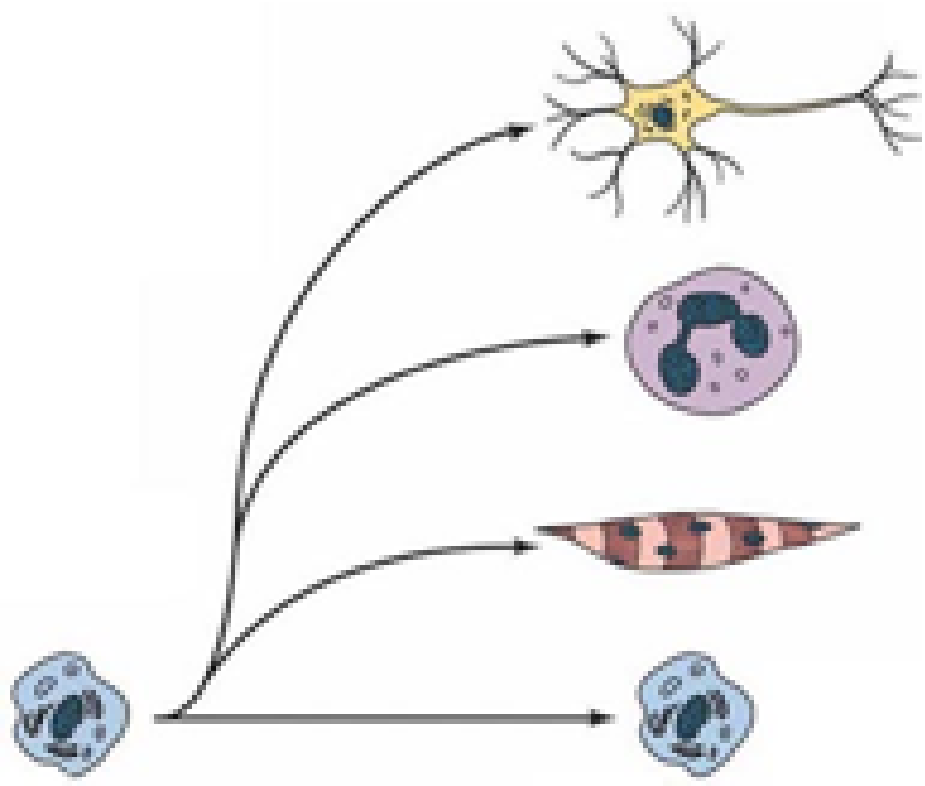

Figure 10.1: Cell differentiation

10.1 Transcription of different classes of genes

RNA-Polymerases(RNAPs) are enzymes that catalyze the synthesis of mRNA during the act of transcription. In eukaryotes there are three classes of RNAPs, designated as I, II, III, that target different classes of genes. RNAPs exhibit high similarity both among the three subclasses and also across different species. High conservation is often a indication of essential biological functionality.

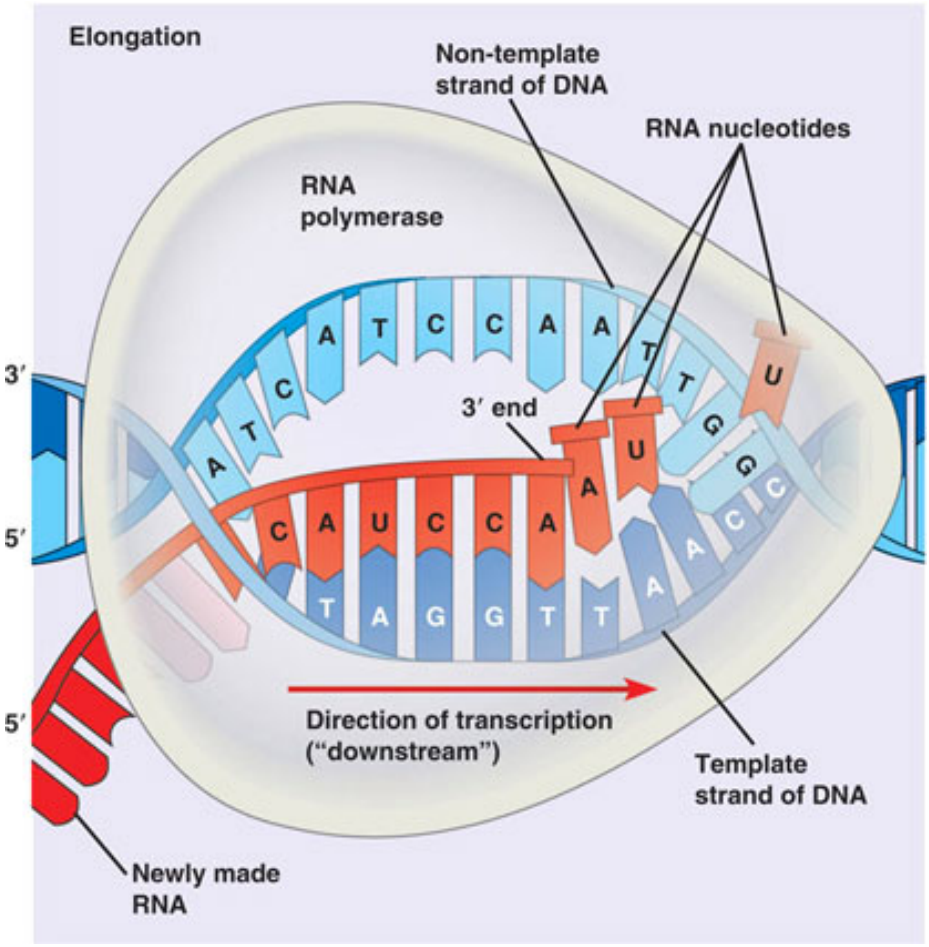

Figure 10.2: Transcription [3]

RNAP II is the main promoter category driven expression of all protein-coded genes.

Its transcription mechanisms have been widely investigated. During transcription, RNAP II slide along the DNA in a 'transcription bubble' of broken up base-pairs. It synthesizes a strand of mRNA that is almost an exact copy of the template DNA but thymine (T) is replaced with uracil (U) as showed in Figure 10.2. The synthesis is also directed in that RNAP II only works in 5' end to 3' end direction.

Transcription involves many transcription factors and a combinatorial array of cis-regulatory DNA elements. Two parts were well characterized for transcription initiation complexes, the *core promoter* and the *co-regulators*. The core promoter is defined as the minimal stretch of contiguous DNA sequence that is sufficient to direct accurate initiation of transcription by the RNA polymerase II.

TATA The TATA box is the first eukaryotic core promoter element to be identified. It is located about 25-30nt upstream of the transcription start site. Its consensus sequence is TATAAA. Contrary to common understanding, that all genes contain a TATA box in their promoters, only 30% of human genes seems to have this. [83].

Inr The Inr element encompasses the TSS and is found in both TATA-containing as well as TATA-less promoters. The consensus sequences is [CT][CT]AN[TA][CT][CT] .

DPE DPE was identified as a downstream promoter binding site from fruit flies TFIID. The DPE is found most commonly in TATA-less promoters. Even though it was mostly studied in fruit flies but it is also present in humans [96] with consensus sequences [AG]G[AT][CT][GAC].

BRE The BRE is a TFII B binding site that is located immediately upstream of some

TATA-boxes that interacts with TFIIB in a sequence-specific manner [53]. Its consensus sequence is [CG][CG][GA]CGCC, where the 3' C in the BRE is followed by the 5' T of the TATA box.

There are three kinds of promoter elements: upstream repressors (URS), upstream activators (UAS), Core Promoter Elements (CPEs) see Figure 10.3). Core promoter elements (CPEs) extends either upstream or downstream for roughly 37bp at the transcription initiation site, which includes BBE (TFIIB recognition element), the TATA box, Inr (initiator), and DPE (downstream promoter element) (see schematic map below 10.4). Bulter and Kadonaga have shown that there are at least four CPEs: BRE, TATA, INR and DPE as showed in Figure 10.4 [25].

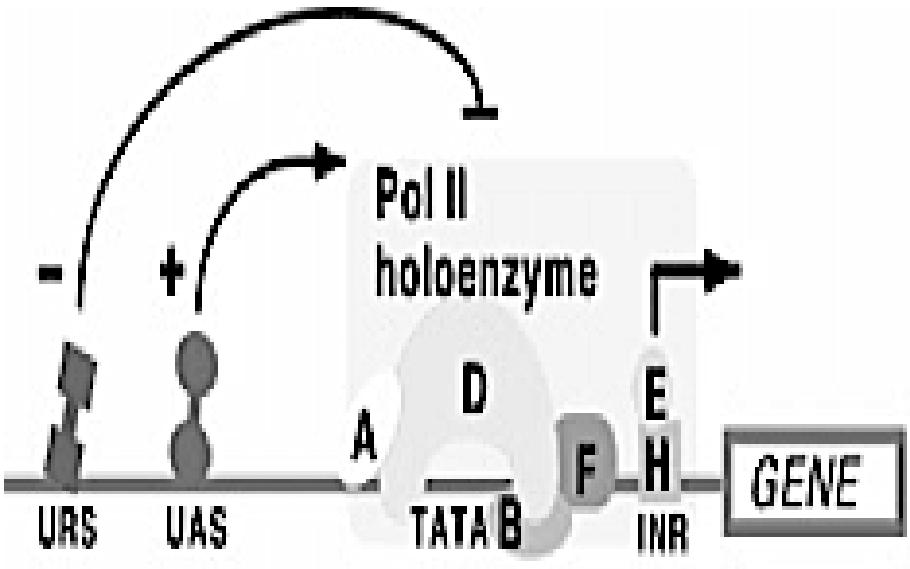

Figure 10.3: Promoter Elements

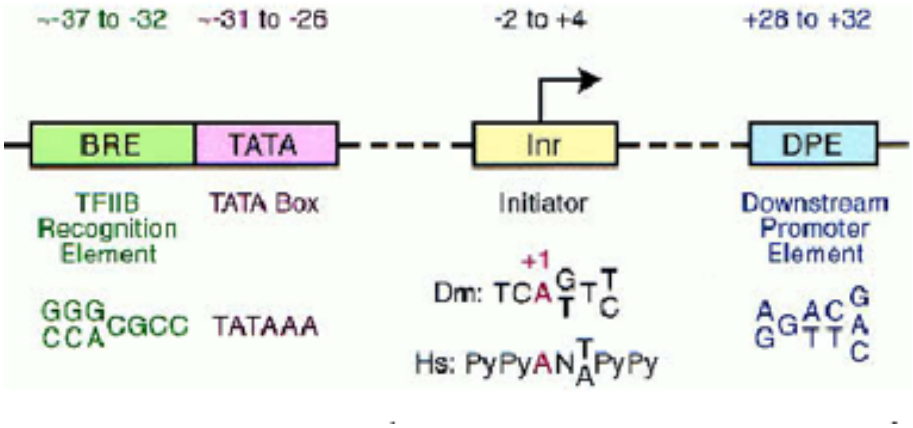

Figure 10.4: Core Promoter Elements [25]

10.2 Current Limitations in the Study of Promoter regions

Despite the critical regulatory role of these non-coding sequences, both experimental and computational methods to identify and predict DNA fragments are extremely limited. Limiting factors include the fact that known CPEs that could characterize promoter regions are small and variable in their sequence composition [4]. Furthermore, because of the high cost and labor-intensive nature of biological experiments, experimental methods to identify binding sites have focused on single genes or small numbers of genes in each experiment, thus data points for inferring binding site classifiers have been sparse. ChIP-chip allows for high resolution genome-wide maps. Although ChIP-chip can be a powerful technique in the area of genomics, it is very expensive. Another limitation is the size of the DNA fragments that can be achieved. Most ChIP-chip protocols utilize sonication as a method of breaking up DNA into small pieces. Antibodies used for ChIP-chip can be an important limiting factor. ChIP-chip requires highly specific antibodies that must recognize its

binding site of the antigen in free solution and also under fixed conditions. A study demonstrating the non-specific nature of DNA-binding proteins has been published that indicates that alternate confirmation of functional relevancy is a necessary step in any ChIP-chip experiment [41, 95].

10.3 Significance of Our Work

Given the availability of high-throughput biological sequence data that localizes promoter regions for hundreds and thousands of genes in the human genome and experimentally verified CPEs patterns, our study systematically computes the relative occurrence of known Core Promoter Elements (CPEs) in promoter regions and evaluates CPEs patterns' prediction power of promoter regions out of the whole genome.

Chapter 11

Research Design and Method

11.1 Data Sets

The project analyzes the relative occurrence of DNA sequence motifs in and around transcription start sites based on three bodies of data. We train our method on CHIP-chip data (Dataset Bing) of two different members of the PolII complex, namely Pol II and TAFII 250 from Dr Ren Bing's lab [71]. The motifs are four PolII core promoter elements (CPEs), termed BRE, TATA, INR, and DPE as described in section 10.

The ENCODE project is sponsored by the National Insititutes of Health (NIH). The pilot project carefully selected 1% the human genome as a true representation of the whole human genome. This was so that it can be used as manageable test data for the algorithms that aimed to annotate the full genomes. We decided to use ENCODE data as the starting point because of the following two reasons:

1. It was well chosen to represent the whole human genome, any finding from this dataset has a lower chance of being some kind of artifact.

2. It is a publicly available dataset with many experimental biologists working with it, thus giving us the opportunity to fine-tune our data analysis later when more experimental data is available.

The Reference Sequence (RefSeq) database [66, 67, 65] is an open access, annotated and curated collection of publicly available nucleotide sequences (DNA, RNA) and their protein products. This database is built by National Center for Biotechnology Information (NCBI) and is the most accurate non-redundant public gene database.

Two subsets from ENCODE [1] are used:

- Set1: 544 fragments overlap with Dataset Bing through querying EnseMart [2] with REGION=Entries in an ENCODE region and GENE= Entries with 5 UTR. We then use BLAT [46] to get the loci of 544 fragments.
- Set2: 494 overlap with RefSeq database at 5' UTR.

We then filtered the CHIP-chip fragments with the requirements that they overlapped with at least one refseqss 5 UTR form Set2. The 91 left over fragments, as shown in Figure 11.1 is defined as true transcription start sites (TSS). These sequences not only have pol II binding sites but also are immediately upstream of a known gene.

11.2 Motif & Super Motif

DNA motifs are small and by random chance, one will find at least one match of a 5 base motif in 1024 (4^5) sequence. Therefore, we need to make sure that our background

[1] http://genome.ucsc.edu/ENCODE/
[2] http://uswest.ensembl.org/biomart/martview/0edc772a1b8f4a70aef48a7bde98e7cb

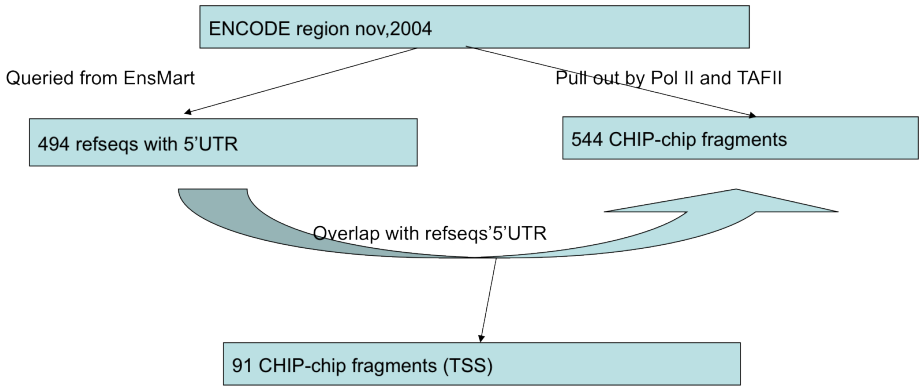

Figure 11.1: Workflow of getting TSS

model has the same length distribution as our TSS dataset ($149bp \sim 941bp$). We also have evidence that GC content is not unified through out the gene model. Thus we made sure that our randomly selected background sequences from the human genome have exactly 41% GC content, which is the average GC content of human genome.

Super Motif: These four core promoter elements have been shown to tend to work together with some distance constraints. There is evidence that if there is a strong binding site on one of them then the other binding site could be relatively weak. In order to capture this kind of binding event, we define a term Super Motif $TATA + Inr$: At least one strong binding sites exists for one of the TATA, Inr motifs. These super-motifs were supplemented with constraints on the number of nucleotides allowed between individual motifs as show in Table 11.1. Given 4 CPEs, there are 15 of these kind of factor combinations:

4 singlton : BRE, $TATA$, Inr and DPE

6 pair : $BRE + TATA$, $BRE + Inr$, $BRE + DPE$, $TATA + Inr$, $TATA + DPE$, $Inr + DPE$

4 triplet : $BRE+TATA+Inr$, $BRE+TATA+DPE$, $BRE+Inr+DPE$, $TATA+Inr + DPE$

1 all four : $BRE + TATA + Inr + DPE$

Pair of CPEs	*Distance in base*
$BRE + TATA$	0
$BRE + INR$	30
$BRE + DPE$	60
$TATA + INR$	26
$TATA + DPE$	54

Table 11.1: Pairwise Distance [25]

11.3 Motif Matching Strategy

Our analysis first searched for matches to the experimentally established consensus motifs for each CPE in Dataset Bing, [GC][GC][GA]CGCC for BRE, TATAAA for TATA, PyPyAN[TA]PyPy for INR, and [AG]G[AT][CT][GAC] for DPE. There is no significant enrichment for any single motif, pair of motifs, or triple of motifs. This indicates that the sequence variation in these binding sites is greater than the variation inherent in the experimentally identified consensus sequences. Moreover, evidence suggests that the DPE and TATA sites specifically do not need to co-occur for a given gene [24].

We then applied a 'fuzzy matching' algorithm through an established probabilistic method of simultaneously searching for binding sites for several transcription factors. The algorithm models the competition of several transcription factors (weight matrices) for a stretch of DNA. This approach should better reflect the biochemistry of protein- DNA interactions than approaches treating the factors independently [94].

$$P = \log \frac{n_i(b) + 1}{N_i + 4}$$

$$b = \{A, T, G, C\} N_i = n_i(A) + n_i(T) + n_i(G) + n_i(C)$$

Because of the lack of knowledge of the weight position matrix, we generated a dummy position weight matrix (PWM) of the core promoter elements based on the motif [25] with pseudo-count 1 to fix the problem of accuracy for our weight matrix. Figure 11.2 shows the PWM of super motif $BRE + TATA$.

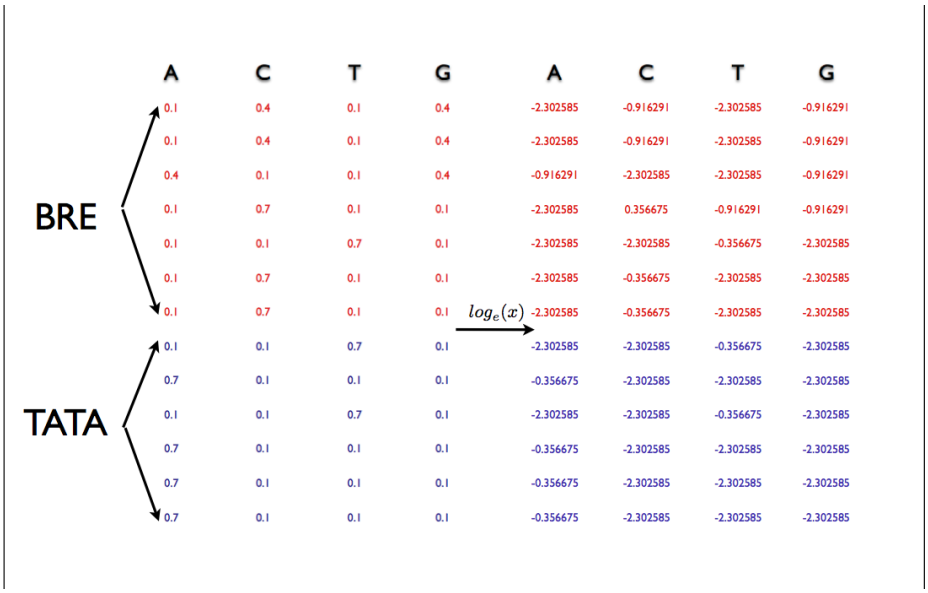

Figure 11.2: PWMs of $BRE + TATA$

Since we do not know how strong the binding sites need to be to enable the transcription initiation, we defined a set of 'infinity' (binding or matching) thresholds to capture the different level of infinity between the factor and binding sites.

We trained our method on the TSS dataset and then applied the method to Dataset Bing. We then confirmed our findings through three unrelated publicly available datasets: DBTSS [3][84] , Mike Snyders CHIP-chip data Dataset Snyder) [58, 73] and Affymetrixs transcription factor dataset (Affymextrix) [27].

[3]http://dbtss.hgc.jp

Chapter 12

Results

12.1 True Transcription Start Site Result

Our initial analysis searched for matches to experimentally established consensus motifs for each CPE in TSS. Although there are enrichments for certain motifs, none of them is statistically significant for single motifs. The sensitivity for TATA was 0.32, BRE was 0.33 INR was 0.13, and DPE was 0.021 as shown in Figure 12.1 or the super motifs from Figure 12.2.

We then compared all 15 factor-combinations on all 8 different infinity-levels between the TSS and background sequences through a relaxed search as described in section 11.3 for motif and super-motifs. Three signals out of 16 were found to be statistically significantly enriched in TSS :BRE in Figure12.3, $INR + DPE$ in Figure 12.4 and $TATA + DPE$ in Figure 12.5.

One interpretation of these results is that for the INR and DPE pair of motifs, one of the two, but not both, is allowed to be relaxed, and similarly for TATA paired with DPE. This interpretation is consistent with experimental results [23, 24].

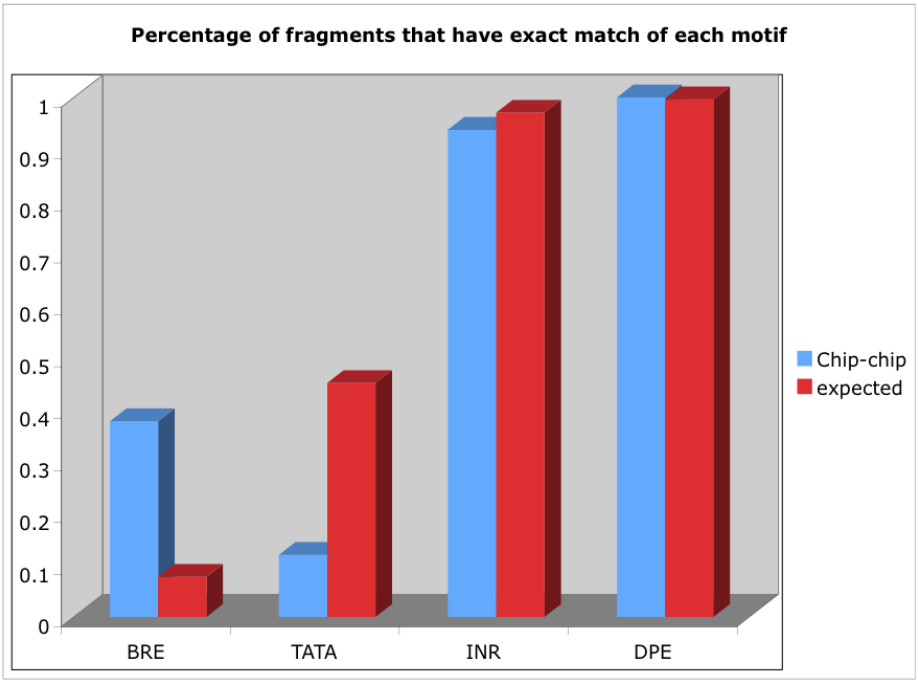

Figure 12.1: Single CPE enrichment

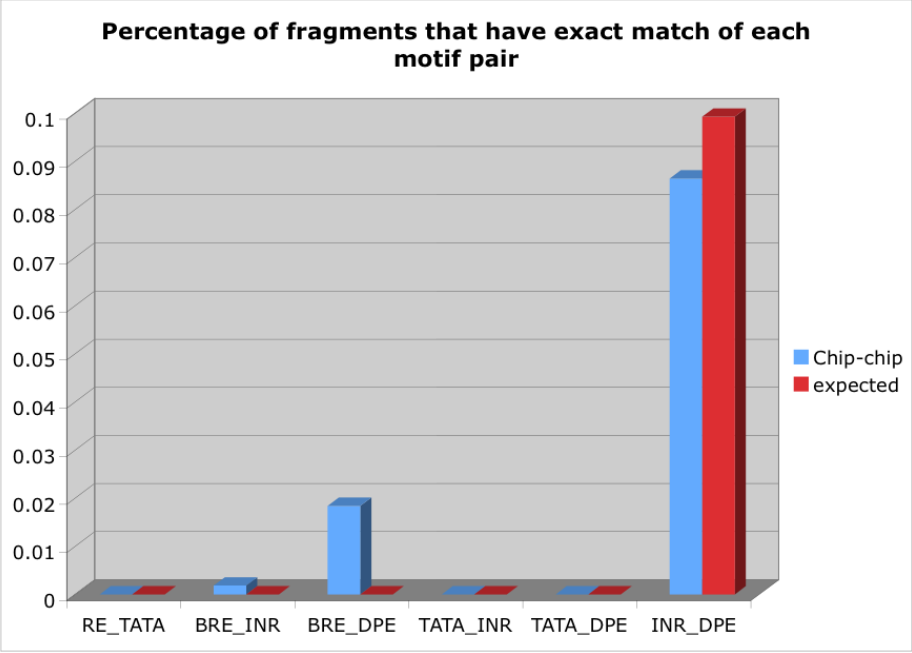

Figure 12.2: Pair CPE enrichment

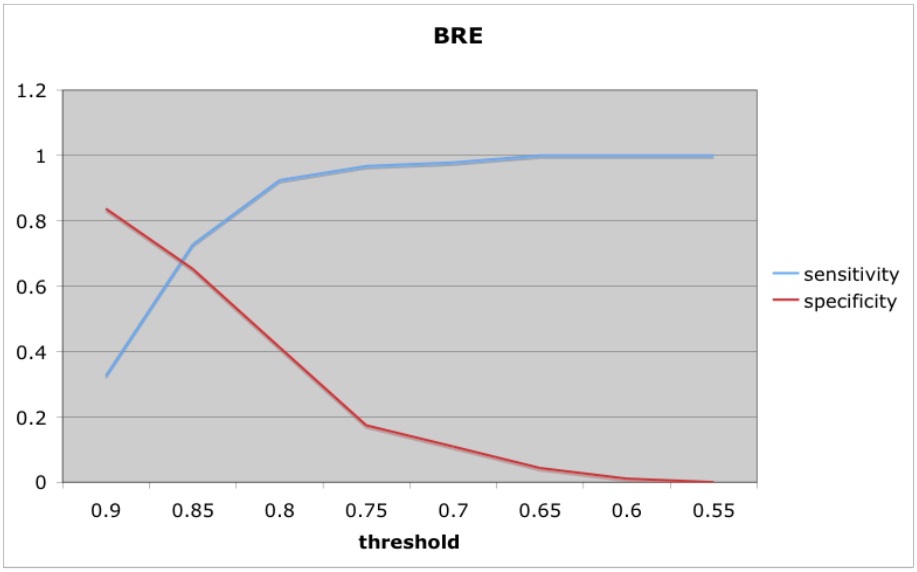

Figure 12.3: Sensitivity and Specificity for BRE

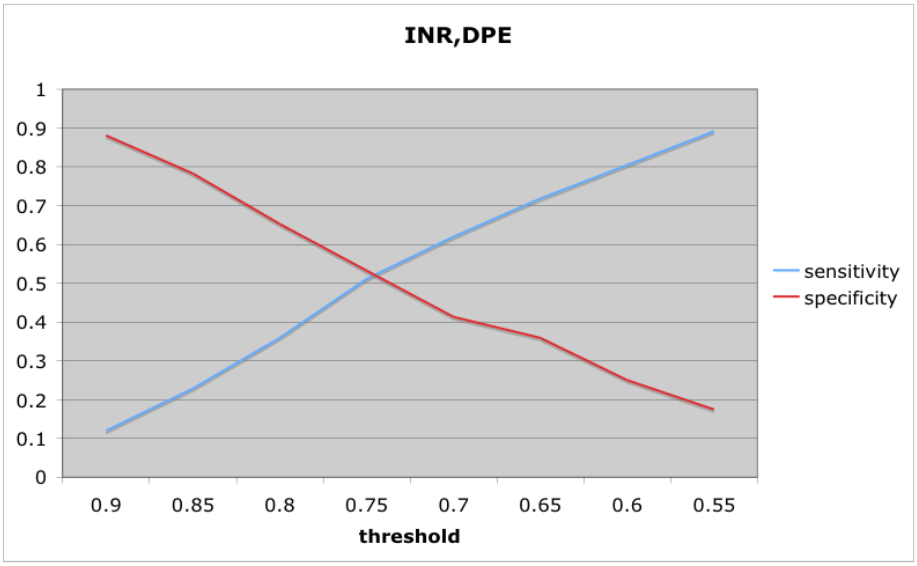

Figure 12.4: Sensitivity and Specificity for Inr and DPE pair

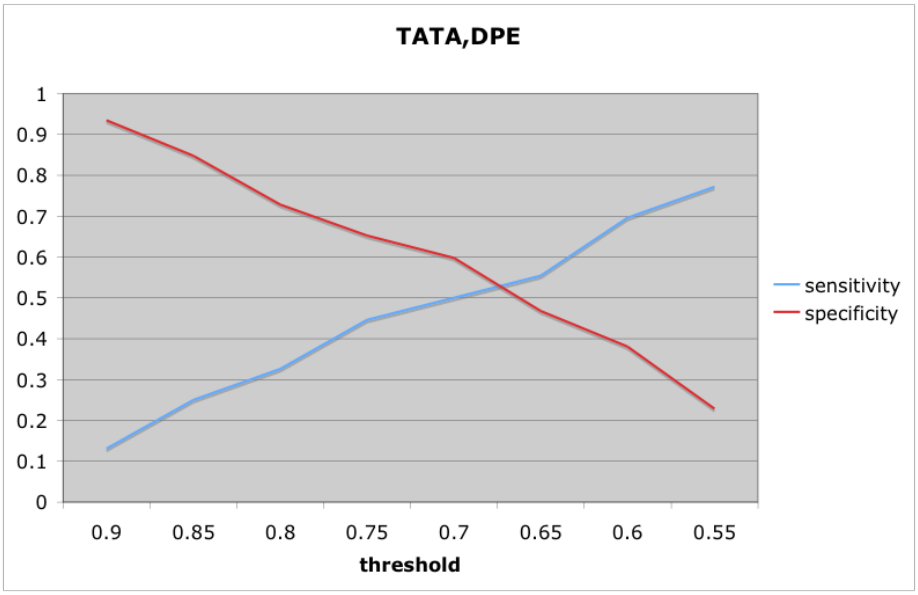

Figure 12.5: Sensitivity and Specificity for TATA and DPE pair

To demonstrate that these signals were not due to experimental artifacts in the CHIP-chip procedure, the whole Dataset Bing and Dataset DBTSS were used for validation, and yielded similar results as our controlled TSS 12.1.

12.2 Other Data Source Results

We extended our validation further to test the $TATA + DPE$ super motif with $infinity = 0.7$ on Dataset Snyder and Dataset Affymetrix. The results were comparable to the Dataset Bing result (Figure12.6), even if the specificities are slightly lower.

These results can be explained by the fact that using the PolII antibody CHIP-chip experiment from Dataset Bing produces a cleaner set of TSS. A similar conclusion can be made through the GC percentage comparison of these datasets, as we can see from the AT/GC frequency plotting of Dataset TSS in Figure 12.7. Thus the high percentage of

544 CHIP-chip data		
motif	sensitivity	specificity
BRE	0.66	0.65
$INR + DPE$	0.70	0.53
$TATA + DPE$	0.70	0.60
12763 DBTSS data		
motif	sensitivity	specificity
BRE	0.80	0.33
$INR + DPE$	0.58	0.42
$TATA + DPE$	0.64	0.55

Table 12.1: DataSet Bing & Dataset DBTSS results

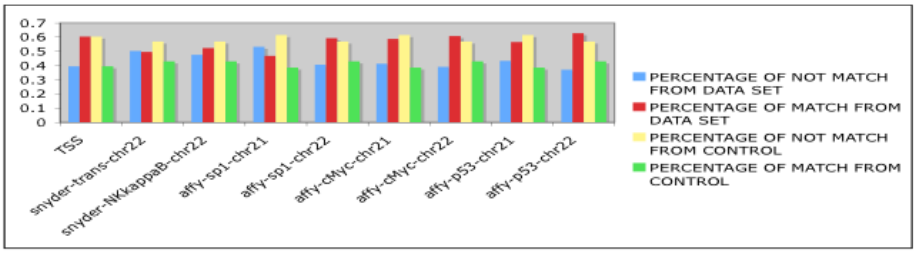

Figure 12.6: Sensitivity and Specificity for Dataset Snyder and Dataset Affymetrix

GC is a good indicator to pinpoint TSS. From Figure 12.8, we can see Dataset Bing's GC% is clearly higher than all the other datasets.

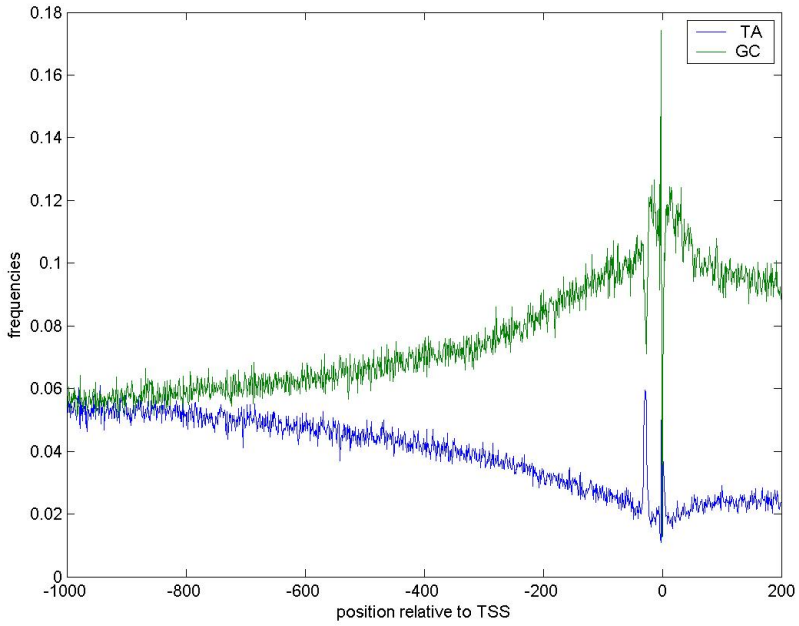

Figure 12.7: AT/GC frequency in TSS region

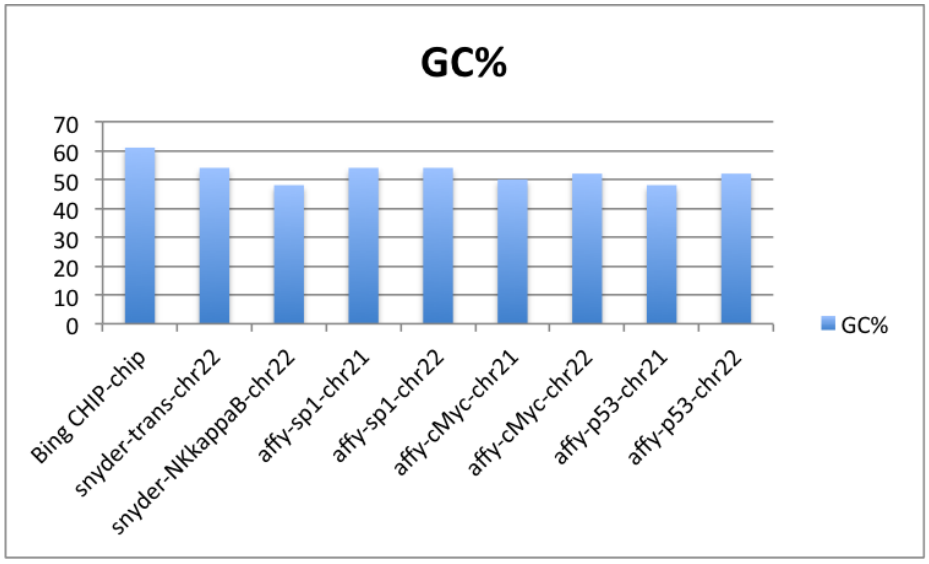

Figure 12.8: GC % of all DataSets

Part IV

Conclusions

Chapter 13

Conclusions and Future Work

We have shown that the n-gram model is a useful technique in summarizing tandem repeats to facilitate clustering. By applying k-means like clustering, we demonstrated that a 3-gram representation is sufficient to capture the content of tandem repeats. We also determined that rank correlation distances outperformed geometric distances for our data. Finally, we have shown through a case study that the classification schema not only refines the original repeat finding algorithm but also improves usability of repeat databases. We have applied our novel classification scheme to the entire dataset of tandem repeats in the human genome of TRedD, and the results will be publicly available through a web application with a database backend at: http//tandem.sci.brooklyn.cuny.edu.

In future work we plan on following up on the idea of using our classification scheme to facilitate the study and comparison of the tandem repeat content on a global scale on publicly available sequencing data such as such as samples from the 1000 Genome Project[1] and The Cancer Genome Atlas (TCGA)[2]

[1] http://www.1000genomes.org/
[2] http://cancergenome.nih.gov/

Our study of CPEs on the CHIP-chip data not only confirmed that most of the time the four core promoter elements do not exist at transcription start sites as has been previously reported by many papers[93, 75, 42], but also reveals that most of the promoters need either a strong TATA element or a DPE elements which has been discovered by Kadonaga et. al. [43]. By extending our analysis on data outside the ENCODE region and data from technology other than ChIP-chip, we confirmed that our finding is true for all promoters in the human genome

In conclusion, sensitivity is dramatically improved by using a relaxed matching method compared to requiring exact matches to motifs. Our results suggest possible signals that could be used to predict transcription start sites when combined with other signals such as CpG islands [31] and clusters of specific transcription binding sites [18]. Our fuzzy search method can be used to assess the degree to which motifs are dependent. Further refinements for the relaxed search will target pairs in which one motif occurs more often in the presence of another motif and rarely alone. For example, BRE is considered to be associated with TATA, but TATA has been observed to occur without BRE.

Bibliography

[1] http://library.thinkquest.org/c0123260/basic%20knowledge/rna/transcription.jpg.

[2] http://upload.wikimedia.org/wikipedia/commons/e/e9/transcription.jpg.

[3] http://www.accessexcellence.org/rc/vl/gg/central.php.

[4] http://www.allisons.org/ll/algds/tree/suffix/.

[5] http://www.ibm.com/developerworks/java/library/j-seqalign/index.html.

[6] http://www.molecularstation.com/molecular-biology-images/data/502/human-genome-gene.png.

[7] http://www.nature.com/nature/journal/v409/n6822/images/409860ab.2.jpg.

[8] H. Akaike. A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723, 1974.

[9] H. Akaike. Likelihood and the bayes procedure. *Trabajos de estadística y de investigación operativa*, 31(1):143–166, 1980.

[10] J. D. U. Alfred V. Aho, John E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[11] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *Journal of Molecular Biology*, 215:403–410, 1990.

[12] O. Aparicio, J. V. Geisberg, E. Sekinger, A. Yang, Z. Moqtaderi, and K. Struhl. Chromatin immunoprecipitation for determining the association of proteins with specific genomic sequences in vivo. *Curr Protoc Mol Biol*, Chapter 21:Unit 21.3, Feb 2005.

[13] O. Aparicio, J. V. Geisberg, and K. Struhl. Chromatin immunoprecipitation for determining the association of proteins with specific genomic sequences in vivo. *Curr Protoc Cell Biol*, Chapter 17:Unit 17.7, Sep 2004.

[14] T. L. Bailey and C. Elkan. Fitting a mixture model by expetation maximization to discover motifs in biopolymers. In *Internationl Conference on Intelligent System for Molecular Biology*, pages 28–36. AAAI Press, 1994.

[15] G. Benson. Tandem repeats finder: a program to analyze dna sequences. *Nucleic acids research*, 27(2):573, 1999.

[16] G. Benson. A new distance measure for comparing sequence profiles based on path lengths along an entropy surface. *Bioinformatics*, 18(suppl 2):S44–S53, 2002.

[17] P. Berkhin. A survey of clustering data mining techniques. In *Grouping multidimensional data*, pages 25–71. Springer, 2006.

[18] B. P. Berman, B. D. Pfeiffer, T. R. Laverty, S. L. Salzberg, G. M. Rubin, M. B. Eisen, and S. E. Celniker. Computational identification of developmental enhancers: conservation and function of transcription factor binding-site clusters in drosophila melanogaster and drosophila pseudoobscura. *Genome biology*, 5(9):R61, 2004.

[19] E. Birney, J. A. Stamatoyannopoulos, A. Dutta, R. Guigó, T. R. Gingeras, E. H. Margulies, Z. Weng, M. Snyder, E. T. Dermitzakis, R. E. Thurman, et al. Identification and analysis of functional elements in 1% of the human genome by the encode pilot project. *Nature*, 447(7146):799–816, 2007.

[20] N. Blow. Dna sequencing: generation next-next. *Nature Methods*, 5(3):267–274, March 2008.

[21] H.-J. Bockenhauer and D. Bongartz. *Algorithmic Aspects of Bioinformatics (Natural Computing Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2007.

[22] T. A. Brown. *Genomes*. Oxford: Wiley-Liss, http://www.ncbi.nlm.nih.gov/books/NBK21128/, 2nd edition, 2002.

[23] T. W. Burke and J. T. Kadonaga. The downstream core promoter element, dpe, is conserved fromdrosophila to humans and is recognized by tafii60 of drosophila. *Genes & development*, 11(22):3020–3031, 1997.

[24] J. E. Butler and J. T. Kadonaga. Enhancer–promoter specificity mediated by dpe or tata core promoter motifs. *Genes & development*, 15(19):2515–2519, 2001.

[25] J. E. F. Butler and J. T. Kadonaga. The rna polymerase ii core promoter: a key component in the regulation of gene expression. *Genes Dev*, 16(20):2583–2592, Oct 2002.

[26] M. F. Carey, C. L. Peterson, and S. T. Smale. Chromatin immunoprecipitation (chip). *Cold Spring Harb Protoc*, 2009(9):pdb.prot5279, Sep 2009.

[27] S. Cawley, S. Bekiranov, H. H. Ng, P. Kapranov, E. A. Sekinger, D. Kampa, A. Piccolboni, V. Sementchenko, J. Cheng, A. J. Williams, et al. Unbiased mapping of transcription factor binding sites along human chromosomes 21 and 22 points to widespread regulation of noncoding rnas. *Cell*, 116(4):499–509, 2004.

[28] E. P. Consortium et al. *PLoS Biol*, 9(4):e1001046, 2011.

[29] C. L. Galindo, L. J. McIver, J. F. McCormick, M. A. Skinner, Y. Xie, R. A. Gelhausen, K. Ng, N. M. Kumar, and H. R. Garner. Global microsatellite content distinguishes humans, primates, animals, and plants. *Mol Biol Evol*, 26(12):2809–2819, Dec 2009.

[30] C. L. Galindo, L. J. McIver, H. Tae, J. F. McCormick, M. A. Skinner, I. Hoeschele, C. M. Lewis, J. D. Minna, D. A. Boothman, and H. R. Garner. Sporadic breast cancer

patients' germline dna exhibit an at-rich microsatellite signature. *Genes, Chromosomes and Cancer*, 50(4):275–283, 2011.

[31] M. Gardiner-Garden and M. Frommer. Cpg islands in vertebrate genomes. *Journal of molecular biology*, 196(2):261–282, 1987.

[32] J. R. Gatchel and H. Y. Zoghbi. Diseases of unstable repeat expansion: mechanisms and common principles. *Nat Rev Genet*, 6(10):743–755, Oct 2005.

[33] Y. Gelfand, A. Rodriguez, and G. Benson. Trdb—the tandem repeats database. *Nucleic acids research*, 35(suppl 1):D80–D87, 2007.

[34] R. Groult, M. Léonard, and L. Mouchard. Speeding up the detection of evolutive tandem repeats. *Theoretical computer science*, 310(1):309–328, 2004.

[35] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge University Press, New York, NY, USA, 1997.

[36] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[37] G. Z. Hertz, I. Hartzell, George W., and G. D. Stormo. Identification of consensus patterns in unaligned DNA sequences known to be functionally related. *Comput. Appl. Biosci.*, 6(2):81–92, 1990.

[38] G. Z. Hertz and G. D. Stormo. Identifying dna and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7):563–577, 1999.

[39] D. S. Hirschberg. A linear space algorithm for computing maximal common subsequences. *Commun. ACM*, 18(6):341–343, 1975.

[40] A. J. Jeffreys. 1992 william allan award address. *American journal of human genetics*, 53(1):1, 1993.

[41] W. E. Johnson, W. Li, C. A. Meyer, R. Gottardo, J. S. Carroll, M. Brown, and X. S. Liu. Model-based analysis of tiling-arrays for chip-chip. *Proc Natl Acad Sci U S A*, 103(33):12457–12462, Aug 2006.

[42] T. Juven-Gershon and J. T. Kadonaga. Regulation of gene expression via the core promoter and the basal transcriptional machinery. *Developmental biology*, 339(2):225, 2010.

[43] J. T. Kadonaga et al. The dpe, a core promoter element for transcription by rna polymerase ii. *Experimental and Molecular Medicine*, 34(4):259–264, 2002.

[44] J. Karkkainen and P. Sanders. Simple linear work suffix array construction. In *In Proc. 30th Internat. Colloq. Automata, Languages and Programming*, pages 943–955, 2003.

[45] L. Kaufman and P. J. Rousseeuw. *Finding groups in data: an introduction to cluster analysis*, volume 344. Wiley-Interscience, 2009.

[46] W. J. Kent. *Genome research*, 12(4):656–664, 2002.

[47] W. J. Kent. Blat-the blast-like alignment tool. *Genome research*, 12(4):656–664, April 2002.

[48] N. R. V. Kiermer. Primer: Sequencing—the next generation. *Nature methods*, 5(15), 2008.

[49] D. K. Kim, J. S. Sim, H. Park, and K. Park. Linear-time construction of suffix array. *Combinatorial Pattern Matching*, pages 186–199, 2003.

[50] P. Ko and S. Aluru. Space efficient linear time construction of suffix arrays. In *In Proc. Fourteenth Annual Symp. Combinatorial Pattern Matching*, pages 200–210, 2003.

[51] R. Kolpakov, G. Bana, and G. Kucherov. mreps: efficient and flexible detection of tandem repeats in dna. *Nucleic acids research*, 31(13):3672–3678, 2003.

[52] P. Kozlowski, M. de Mezer, and W. J. Krzyzosiak. Trinucleotide repeats in human genome and exome. *Nucleic acids research*, 38(12):4027–4039, 2010.

[53] T. Lagrange, A. N. Kapanidis, H. Tang, D. Reinberg, and R. H. Ebright. New core promoter element in rna polymerase ii-dependent transcription: sequence-specific dna binding by transcription factor iib. *Genes & development*, 12(1):34–44, 1998.

[54] E. S. Lander, L. M. Linton, B. Birren, C. Nusbaum, M. C. Zody, J. Baldwin, K. Devon, K. Dewar, M. Doyle, W. FitzHugh, et al. Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921, 2001.

[55] M. Li, B. Ma, and L. Wang. Finding similar regions in many sequences. *J. Comput. Syst. Sci.*, 65(1):73–96, 2002.

[56] S. R. Maetschke, K. S. Kassahn, J. A. Dunn, S.-P. Han, E. Z. Curley, K. J. Stacey, and M. A. Ragan. A visual framework for sequence analysis using n-grams and spectral rearrangement. *Bioinformatics*, 26(6):737–744, 2010.

[57] U. Manber and Myers. Suffix arrays: a new method for on-line string searches. *SIAM Journal on Computing*, 22:935–948, 1993.

[58] R. Martone, G. Euskirchen, P. Bertone, S. Hartman, T. E. Royce, N. M. Luscombe, J. L. Rinn, F. K. Nelson, P. Miller, M. Gerstein, S. Weissman, and M. Snyder. Distribution of nf-kappab-binding sites across human chromosome 22. *Proc Natl Acad Sci U S A*, 100(21):12247–12252, Oct 2003.

[59] E. McCreight. A space-economical suffix tree construction algorithm. *J. of the ACM*, 23, 1976.

[60] S. M. Mirkin. Dna structures, repeat expansions and human hereditary disorders. *Curr Opin Struct Biol*, 16(3):351–358, Jun 2006.

[61] A. Moses, D. Chiang, and M. Eisen. Phylogenetic motif detection by expectation maximaization on evolution mixtures. In *Pac Symo Biocomput*, pages 324–335, 2004.

[62] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, March 1970.

[63] D. Pelleg, A. Moore, et al. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the seventeenth international conference on machine learning*, volume 1, pages 727–734. San Francisco, 2000.

[64] M. Pellegrini, M. E. Renda, and A. Vecchio. Tandem repeats discovery service (treads) applied to finding novel cis-acting factors in repeat expansion diseases. *BMC bioinformatics*, 13(Suppl 4):S3, 2012.

[65] K. D. Pruitt, T. Tatusova, W. Klimke, and D. R. Maglott. Ncbi reference sequences: current status, policy and new initiatives. *Nucleic Acids Res*, 37(Database issue):D32–D36, Jan 2009.

[66] K. D. Pruitt, T. Tatusova, and D. R. Maglott. Ncbi reference sequence (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*, 33(Database issue):D501–D504, Jan 2005.

[67] K. D. Pruitt, T. Tatusova, and D. R. Maglott. Ncbi reference sequences (refseq): a curated non-redundant sequence database of genomes, transcripts and proteins. *Nucleic Acids Res*, 35(Database issue):D61–D65, Jan 2007.

[68] M. A. Ptashne. Genes & signals. 2001.

[69] A. R. Quinlan and I. M. Hall. Bedtools: a flexible suite of utilities for comparing genomic features. *Bioinformatics*, 26(6):841–842, 2010.

[70] S. Rao, A. Rodriguez, and G. Benson. Evaluating distance functions for clustering tandem repeats. *GENOME INFORMATICS SERIES*, 16(1):3, 2005.

[71] B. Ren, B. D. Dynlacht, et al. Use of chromatin immunoprecipitation assays in genome-wide location analysis of mammalian transcription factors. *Methods in enzymology*, 376:304, 2004.

[72] B. Ren, F. Robert, J. J. Wyrick, O. Aparicio, E. G. Jennings, I. Simon, J. Zeitlinger, J. Schreiber, N. Hannett, E. Kanin, et al. Genome-wide location and function of dna binding proteins. *Science Signaling*, 290(5500):2306, 2000.

[73] J. L. Rinn, G. Euskirchen, P. Bertone, R. Martone, N. M. Luscombe, S. Hartman, P. M. Harrison, F. K. Nelson, P. Miller, M. Gerstein, et al. The transcriptional activity of human chromosome 22. *Genes & development*, 17(4):529–540, 2003.

[74] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[75] A. Sandelin, P. Carninci, B. Lenhard, J. Ponjavic, Y. Hayashizaki, and D. A. Hume. Mammalian rna polymerase ii core promoters: insights from genome-wide studies. *Nature Reviews Genetics*, 8(6):424–436, 2007.

[76] G. Schwarz. Estimating the dimension of a model. *The annals of statistics*, 6(2):461–464, 1978.

[77] C. E. Shannon and W. Weaver. A mathematical theory of communication, 1948.

[78] J. Shendure and H. Ji. Next-generation dna sequencing. *Nature Biotechnology*, 26(10):1135–1145, October 2008.

[79] S. Sinha and M. Tompa. Ymf: a program for discovery of novel transcription factor binding sites by statistical overrepresentation. *Nucleic Acids Research*, 31(13):3586–3588, 2003.

[80] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, March 1981.

[81] D. Sokol and F. Atagun. Tredd—a database for tandem repeats over the edit distance. *Database: The Journal of Biological Databases and Curation*, 2010, 2010.

[82] D. Sokol, G. Benson, and J. Tojeira. Tandem repeats over the edit distance. *Bioinformatics*, 23(2):e30–e35, 2007.

[83] Y. Suzuki, T. Tsunoda, J. Sese, H. Taira, J. Mizushima-Sugano, H. Hata, T. Ota, T. Isogai, T. Tanaka, Y. Nakamura, et al. Identification and characterization of the potential promoter regions of 1031 kinds of human genes. *Genome research*, 11(5):677–684, 2001.

[84] Y. Suzuki, R. Yamashita, K. Nakai, and S. Sugano. Dbtss: Database of human transcriptional start sites and full-length cdnas. *Nucleic acids research*, 30(1):328–331, 2002.

[85] E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14, 1995.

[86] J. C. Venter, M. D. Adams, E. W. Myers, P. W. Li, R. J. Mural, G. G. Sutton, H. O. Smith, M. Yandell, C. A. Evans, R. A. Holt, et al. The sequence of the human genome. *Science Signaling*, 291(5507):1304, 2001.

[87] Z. Volkovich, V. Kirzhner, A. Bolshoy, E. Nevo, and A. Korol. The method of n-grams in large-scale clustering of dna texts. *Pattern recognition*, 38(11):1902–1912, 2005.

[88] P. Weiner. Linear pattern matching algorithms. In *IEEE Conference Record of 14th Annual Symposium on Switching and Automata Theory SWAT '08*, pages 1–11, Oct. 15-17, 1973.

[89] Y. Wexler, Z. Yakhini, Y. Kashi, and D. Geiger. Finding approximate tandem repeats in genomic sequences. *Journal of Computational Biology*, 12(7):928–942, 2005.

[90] R. Xu, D. Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[91] R. Xu, D. Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[92] R. Xu and D. C. Wunsch. Clustering algorithms in biomedical research: a review. *Biomedical Engineering, IEEE Reviews in*, 3:120–154, 2010.

[93] C. Yang, E. Bolotin, T. Jiang, F. M. Sladek, and E. Martinez. Prevalence of the initiator over the tata box in human and yeast genes and identification of dna motifs enriched in human tata-less core promoters. *Gene*, 389(1):52, 2007.

[94] M. Zavolan, N. Rajewsky, N. D. Socci, and T. Gaasterlamd. Smashing regulatory sites in dna by human-mouse sequence comparisons. In *Bioinformatics Conference, 2003. CSB 2003. Proceedings of the 2003 IEEE*, pages 277–286. IEEE, 2003.

[95] M. Zheng, L. O. Barrera, B. Ren, and Y. N. Wu. Chip-chip: data, model, and analysis. *Biometrics*, 63(3):787–796, Sep 2007.

[96] T. Zhou and C.-M. Chiang. The intronless and tata-less humantaf ii 55 gene contains a functional initiator and a downstream promoter element. *Journal of Biological Chemistry*, 276(27):25503–25511, 2001.