Jian Xu

Building a dictionary for DNA
Decoding the regulatory regions of a genome

Harmen J. Bussemaker, Hao Li, and Eric D. Siggia
PNAS Aug. 29, 2000, vol 97, p 10096-10100

More than 95 percent of DNA is called "Junk DNA" by molecular biologists, because they are unable to ascribe any function to it. However it has been found that the sequence of the syllables is not random at all and has a striking resemblance with the structure of human language [1]. Therefore, scientists now generally believe that this DNA must contain some kind of coded information. But the code and its function is yet completely unknown. It has been speculated that this region of DNA may contribute to the cellular processes such as regulation of transcription. Therefore, deciphering the information coded in the regulatory regions may be critical to the understanding of transcription in a genomic scale. Yet the development of computational tools for identifying regulatory elements has lagged behind those for sequence comparison and gene discovery.

Former approaches to decipher regulatory regions use 10~100 coregulated genes and then find a pattern common to most of the upstream regions[2,3]. Analysis tools range from general multiple alignment algorithms to comparison of the frequency counts of substrings with some reference set[3]. These approaches typically reveal a few responsive elements. In Harmen et. al.'s paper, they use statistical analysis to build a most probable dictionary of words and motifs for the DNA regulatory region. Their algorithm "MobyDick" is suitable for discovering motifs that are responsible for regulatory process.

Theory and Methods

The motifs in regulatory regions in eukaryotic genomes are typically separated by random spacers that have diverged sufficiently to be modeled as a random background (Fig.1). The sequence data are modeled as the concatenation of words *w* drawn at random with frequency p_w from a probabilistic "dictionary" D; and there is no syntax. Therefore, the likelihood function Z(S, p_w), that is, the probability of obtaining a sequence S for a given normalized probabilistic dictionary {p_w} is:

$$Z(S, p_w) = \sum_P \prod_w (p_w)^{N_w(P)} \qquad (1)$$

where the sum is over all possible segmentations P of S, i.e. all possible ways to divide S into the words in the dictionary. N_w (P) is the number of times the word *w* is used in a segmentation P. For example, given a probabilistic dictionary

$$\begin{Bmatrix} A -> p_A \\ T -> p_T \\ AT -> p_{AT} \end{Bmatrix}$$ (word *w* -> possibility p_w),

and a sequence S=TATA, there are two possible segmentations: P_1=T.A.T.A, and P_2=T.AT.A, where "." Denotes a word separator. Thus the possibility of obtaining sequence TATA in this dictionary

$$Z(TATA, p_w) = p_A^2 p_T^2 + p_A p_T p_{AT}$$

The dictionary of sequence S is constructed by iterating the following two steps:

1. Fitting step: given words in the dictionary, find $\{p_w\}$ by maximizing the likelihood function $Z(S,p_w)$.
2. Adding new words: do statistical test on longer words based on the current dictionary, add the ones that are over-represented to the dictionary. Then go back to step 1 to re-assign $\{p_w\}$.

In the first step, in order to maximize the likelihood function Z, they defined a "free energy" f= -ln(Z) / L, and "energy" p_w= exp(-E_w), where L is the total length of the sequence. Therefore, finding $\{p_w\}$ to maximize the likelihood function Z with the constraint $p_w \geq 0 \; and \; \sum p_w = 1$ is equivalent to finding parameters which minimizes the "free energy", that is, solving for p_w from equation

$$p_w = < N_w > / \sum_{w'} < N_{w'} > \qquad (2)$$

where $< N_w >= p_w \frac{\partial}{\partial p_w} \ln Z$ is the average number of words w in the ensemble defined by Z. By using a dynamic programming-like technique, they can calculate Z and its various derivatives (up to the second order) in $O(LDl)$, where D is the dictionary size, and l is the maximum word length.

In the second step, they do statistical tests on longer words based on their predicted frequencies from the current dictionary. For example, if the current dictionary D is {A,T,AT}, then the expected frequency of the length-3 word TAT is the sum of frequency of AT.A.T, T.A.T, and T.AT. Then they check whether the average number of occurrences of the composite word created by juxtaposition exceeds a statistically significant. Since the statistical significance of longer words is based on the probability of shorter words, this method does not need an external reference data set to define probability. But the juxtaposition method will miss the words that are not built from fragments that are already exists in the dictionary. In order to overcome this problem, they designed routines to search for over-represented motifs exhaustively within certain classes, including up to two International Union of Pure and Applied Chemistry (IUPAC) symbols representing the 12 distinct subsets of two or more bases. In addition, they also search for motifs consisting of two short strings separated by a gap, or so-called dimmers.

The dictionary is built after iterating step 1 and 2. But that does not permit one to "read" the text in a unique way, because the decomposition into words is probabilistic, so there are many ways to segment the data into words. In order to solve this problem, they let n_w be the number of matches of the string w anywhere in the sequence, and $<N_w>$ be the average number of times the string w is delimited as a word among all segmentations of the data (Fig.2). The ratio $<N_w>/n_w$ serve as a quality factor Q_w. When Q_m is close to

one, almost all occurrences of the string *w* are attributed to the word *w* itself, thus *w* can be clearly delineated from the background.

Results

To test their algorithm, they first applied it to English text, the novel *Moby Dick*. They took the first 10 chapters (about 10^5 characters), reduced all characters to lowercase string containing no spaces or other punctuation characters, and inserted random background (Fig.3). The starting dictionary was just the English alphabet. Words are added by juxtaposition only and were required to have at least two copies in the data. The original text has 4,214 unique words, and 1,600 of them have more than one copy. Their final dictionary had 2,450 words; among the most significant 1,050 were 700 English words and 40 composite words, which is satisfactory.

After the first test, they applied it to all the upstream regions in the yeast genome. They define the regulatory regions to extend upstream from the translation start site to the next coding region but no more than 600 bp. Starting from a single-base dictionary, words were added via an exhaustive search procedure. The final dictionary had 1,200 words. On average, two thirds of the sequence data were segmented into single-letter words and an additional 15% into words of length 4 or less. About 500 words fell above a plausible significance level as defined by the quality factor. These included good matches to about half of the motifs found in ref. [3],[4],[5]. Some of the results are shown in Table 1. This verifies the validity of the algorithm, and shows its capability to discovery new regulatory sites.

As a result, their algorithm does not need to group genes, does not need external reference data set, and can handle large data set ($\sim 10^7$) and many motifs ($\sim 10^3$). Furthermore, their algorithm can be generalized to handle fuzzy motifs. Therefore, it may be a reasonable approach to the identification of regulatory sites.

ATTGTATATAAAGTGTACGCGTGACC

Fig.1. An example of the motifs in a regulatory region. The underlined letters correspond to a motif, the letters between can be deemed as random background.

Fig.2. An example of the quality factor. String "the" occurs two time in this sequence, hence $_{\rightarrow the}$=2. But only one of them is counted as a words, another one is the concatenation of "visit" and "here", thus N_{the}=1. Therefore, quality factor $Q_{the}=N_{the}/_{\rightarrow the}=1/2$.

Moby Dick: CHAPTER 1
Loomings

Call me Ishmael. Some years ago- never mind how long precisely-having little or no money in my purse, and nothing particular to interest me on shore, I thought I would sail about a little and see the watery part of the world. It is a way I have of

(a)

chapterptgpbqdrftezptqtasctmvivwpecjsnisrmbtqlmlfvetl
loomingsfkicallxjgkmekysjerishmaeljplfsomeylqyearstvh
njbagoaxhjtjcokhvneverpmqpmindhowzrbdlzjllonggbhqi
preciselysunpvskepfdjktcgarwtnxybgcvdjfbnohavinglittl

(b)

Fig.3. An analog of DNA sequence by using English text. (a) The original text of the novel *Moby Dick*. (b) The sequence after reducing all characters to lower cases, eliminating all blank or punctuation characters, and inserting random characters as random background. This sequence serves as the input of the algorithm.

Table 1. Known cell cycle sites and some metabolic sites that match words from our genomewide dictionary

MCB	ACGCGT	AAACGCGT ACGCGTCGCGT CGCGACGCGT TGACGCGT
SCB	CRCGAAA	ACGCGAAA
SCB'	ACRMSAAA	ACGCGAAA ACGCCAAA AACGCCAA
Swi5	RRCCAGCR	GCCAGCG GCAGCCAG
SIC1	GCSCRGC	GCCCAGCC CCGCGCGG
MCM1	TTWCCYAAWNNGGWAA	TTTCCNNNNNNGGAAA
NIT	GATAAT	TGATAATG
MET	TCACGTG	RTCACGTG TCACGTGM CACGTGAC CACGTGCT
PDR	TCCGCGGA	TCCGCGG
HAP	CCAAY	AACCCAAC
MIG1	KANWWWWATSYGGGGW	TATATGTG CATATATG GTGGGGAG
GAL4	$CGGN_{11}CCG$	$CGGN_{11}CCG$

The strings in the dictionary words that match the consensus sequence are underlined.

---

[1] Flam, F. "Hints of a language in junk DNA", Science 266:1320, 1994
[2] Eisen, M. B., et al., (1998) Proc. Natl. Acad. Sci. USA 95, 14863-14868.
[3] van Helden, J., Andre, B. & Collado-Vides, J. (1998) J. Mol. Biol. 281, 827-842.
[4] Spellman, P. T., et al., (1998) Mol. Biol. Cell. 9, 3273-3297.
[5] Chu, S., et al. (1998) Science 282, 699-705.