

Chemistry 260 Term Paper
Simon J Galbraith
6/7/2001

Harmen J. Bussemaker, Hao Li, and Eric D. Siggia.

Building a dictionary for genomes: Identification of presumptive regulatory sites by statistical analysis.

Bussemaker developed a statistical algorithm to identify DNA sequence motifs that control gene expression. The algorithm, Moby Dick, decomposes a set of DNA sequences into the most probable dictionary of motifs (words). They construct the most probable word set by segmentation by constructing long words from the frequency of shorter ones. This assumes that a word is composed completely of smaller pre-existing words identified by the algorithm. This approach is separate from research to identify patterns in the upstream regions by multiple alignment algorithms and frequency counting techniques.

The second assumption is that gene regulation is caused by a combinatorial number of regulatory elements, and this requires whole genome analysis to identify potential candidates. An explicit assumption in the work is that regulatory motifs evolved from combinations of smaller sequences.

Moby Dick models sequence data as the concatenation of words (w) that have been drawn at random with a particular frequency (f). In this model words can be variable length as long as they are composed of existing dictionary words. We extract individual words as follows:

1. Start with frequency of individual letters.
2. Find over represented pairs, determine their probability, and add to dictionary.
3. Compute optimal f for each word in dictionary
4. Predict new possible words and add to dictionary: go to step 3.
5. Terminate when no w is above a predetermined threshold.

Given a set W of words (w), they calculate the optimal f_w by maximizing the probability of obtaining the sequence S from the normalized p of all words:

$$Z(\text{sequence}, f_w)$$

In the paper:

$$Z = \sum_p \prod_w [f_w^{\text{(number of times } w \text{ used in a given segmentation)}}]$$

A segmentation is composed from the words in the dictionary; it is how we compose the word from the dictionary. It is best to think of the set of segmentations of a word w being all possible compositions of the w from the dictionary. The above equation means we

want to compose a word in terms of maximal probability, and when these converge we have found the maximal dictionary representative of the sequence. They fit f_w to the sequence, S , by maximizing Z where $\sum_w f_w = 1$ and $f_w \geq 0$.

So what is this really doing? First we want to avoid representing S by itself, and we want to avoid representing S by highly frequent but non-unique w because such w 's are composed by overlap – what they term random juxtaposition. We build the dictionary by finding the best fit of S by the lexicon (calculating f_w and $f_{w'}$ where the word is extended by 1 nucleotide. This process is slow, and they use a series of transformations to speed it up and converge to the global maximum for the sequence. This is an interesting approach as they are building up global signal from a local context in parallel. However, they are not approaching the true global signal. This is evident in the fact that they had to cluster similar oligonucleotides. The obvious answer is that they are over representing the signal in the dictionary, yet the algorithm does not capture the signal directly as the true consensus motif. This is indicated by the need to cluster similar sequences. This is indicative of the mismatch problem in which high identity motifs exist as multiple dictionary entries, which should be merged. Considering the consensus sequence in such cases and calculating its probability is important. It is the consensus sequence that keeps track of the multiple spellings of the regulatory motif. Because of the nature of possible protein interactions: loose binding versus strict binding it makes sense that regulatory motif not be defined by an all or nothing sequence; variability is key.

Moby Dick cannot find any motifs that are not composed from pairs pre-existing in the dictionary. They solve this problem by searching for over represented words by recreating the sequence S from all possible dictionary combinations and counted the frequency of each string. They calculate a probability for each string (Z score) and have a threshold to add it to the dictionary. They also searched for dimers separated by a gap of 3-30 bases. Motifs with gaps larger than 30 bases are missed and n -mers with gaps that contain overlapping motifs are missed.

Motifs that overlap are not identified. The biological example of this is that proteins bind next to each other in one gene but diverged in another. Do we include in the dictionary from both motif A and motif B or simply take the concatenation of AB? This situation may arise that from paralogous duplication and divergence events where we might find AB (low frequency) but not A and B. This should also be seen in orthologous events, and for this reason the current gold standard is to find conserved regulatory regions across multiple genomes.

At the same time we should see a clustering of motifs in particular regions. This is because multiple binding motif binding proteins interact with each other. The assumption is that the spacing between motifs is not random because sets of co-regulating motifs group together for protein-protein interactions that effect transcription. Down regulation, for example, might be accomplished by transcriptional interference where RNA polymerase II is prevented from binding to the start site. Therefore, if we also look at the relative distances between identified regulatory motifs we should be able to see this phenomenon.

They applied Moby Dick to a set of English text, the yeast genome, and a set of 500 genes that are up regulated during sporulation. From the English text of 4,214 unique words of which 2,630 were unique, Moby Dick created a dictionary of 3,600 words. Of these 1,800 were significant (high quality factor) and they found 800 English words, 800 concatenations of English words and 200 word fragments. Any word that had multiple copies occurred in the dictionary. The question remaining, which is not answered in the paper, is that of the 800 concatenations, how many are composed from the 800 English words found? We will see later that such juxtaposition might have biological relevance, and the clustering of such words is important.

Checking against an English text is a good validation. An interesting test would be to run the algorithm against several words that have a single base mutation. For example: the American and British spellings of: tire and tyre respectively. Both have the same functional meaning (a wheel on a car) but have different spelling. However, would both be incorporated into the dictionary? In such cases, biologically we assume that high identity sequences have similar function. After all we are looking for conservation to find regulatory motifs.

In principle, motif detection in a genome is harder to validate because we do not know the complete set of valid lexes. Not only is discovering the dictionary more probable from random chance, but we have a smaller starting alphabet from which to create the words. This increases the number of potential false positives we can create. Consider some primitive production rules: insertion, deletion, and substitution. We can compose many sequences this way. In contrast, the English language has an understood set of production rules so we can easily filter words that do not make sense. More importantly, words in the English language have a fixed spacing requirement, and because of this we limit the types of composition of words by other words. Words are more likely to be composed from smaller words, and in most cases just the alphabet. This is an important consideration, and it would be interesting to analyze the words not detected by the algorithm and determine if, for example, are represented by smaller more frequent (sub) words. This problem may be magnified in a genomic dictionary especially when considering variable length gaps.

Bussemaker ran the algorithm on gene control regions defined as extending upstream from initiation start site to the next coding region on either strand to a limit of 600 base pairs. They removed exact repeats longer than 16 base pairs and ignored fragments less than 100 base pairs. The dictionary had 1,200 words of which 100 were similar oligonucleotides. 500 motifs had high quality factors, which matched well to the MCB, SCB, and MCM1 cell-cycle motifs. They also found roughly 400 dimers (motifs separated by a 3-30 nucleotide gap) that they clustered together. Matched to the database of yeast motifs they found 114 out of 443 non-redundant experimentally confirmed sites (25%).

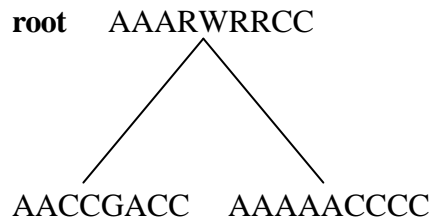
If we take a set of co-regulated genes, and the set of upstream regions in the genome and ask the question are conserved upstream and intronic sequences regulatory, we have to answer two meaningful biological questions. Firstly, how do we deal with the mismatch

problem (finding the optimal superset word), and secondly also consider regulatory motif location. Answering these questions should improve the Moby Dick algorithm.

The mismatch problem is apparent from the need to cluster detected motifs. This is an over representation in the dictionary of the true regulatory motif. An initial improvement to the algorithm would be generating the consensus motif immediately and scoring a probability for it. Although this appears at the outset to be quite tricky we can easily see that as we generate words in the dictionary we should keep a vector of all possible consensus sequences for a given detected word. In the simple case of single nucleotides we would have: A, T, C, G all with the R (A/G), W (C/G) and N (any nucleotide) consensus sequences. One could envisage this vector being stored as a dynamically growing linked-list or vector (dynamic array) data structure.

From this we would generate a tree of sequences where the root node would be the ultimate consensus and the leaves of the tree are motifs detected by the algorithm. We could have multiple (sub) consensuses at each convergence of the tree (a parent node). Each leaf sequence would have an evidence weight, the probability that it contributes to the consensus parent node (say the dictionary probability, or the frequency probability) and as we progress from the leaves to the root node, the higher levels would have a calculated probability based on its children. This would simply look like an AND tree.

Tree 1: Building a Consensus



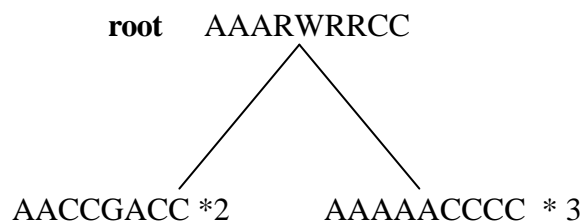
Generating the root node would require a recursive definition that could take place in parallel as we identify words from our sequence. This is a sharp contrast from the problem of identifying words in the English language. This utilizes the biological notion that regulatory motif sites are not necessarily exact and the real region is variable. Beyond the data structure the next essential part is to determine the consensus sequence and calculate how likely it is (posterior odds) by analyzing the maximal contributed evidence from its children. From original dictionary probabilities and perhaps frequency scores or weighted matrices (however we determine the original evidence), we can traverse up the tree in parallel (left side, right side) and calculate probabilities.

In fact in what I've termed the AND tree, a node itself is only conditionally dependent on the right child AND the left child. This property of the tree defines the tree as a second order Markov chain. In fact it may be the case that the left tree contributes more weight than the right tree (based on evidence). Such weights would naturally be represented well as emission probabilities. This answers the question, how likely is it that the parent node emitted the children. In fact what we want is the optimal parent node that emits both children. Then the question is, how do we maximize the joint probability of two

child nodes to generate a sequence, which would emit them? If the right child motif is more probable (i.e. we see it more in the data) then it should contribute more to the consensus parent. Consequently, we need a method to calculate the relative contributions of both trees (they may not be equal). In fact should we generate such trees only when the contributing probabilities are equivalent? Well the answer is not so straightforward.

Biologically, the true regulatory motif is hidden due to the variable nature of the binding sites. For this reason we want to consider the evidence from each child independently and generate the consensus based on each contribution. We want the consensus that maximizes the observations. Consider again the example in the above tree.

Tree 2: Building a Consensus with Evidence



Say for example, we have seen two copies of AACCGGCC and three copies of AAAAACCCC. We calculate the log probability of the root consensus as the sum of the log probabilities of its children. These are, in turn, built up from the left and right children. The probability of the terminal leaves can be determined either from their frequency in the sequence, or from a dictionary built approach as described by Bussemaker. Ideally, we want to consider all possible permutations and combinations of words built from the original sequence. This combinatorial approach can be made more tractable by searching for control elements only in upstream gene and intronic sequence. In this example our consensus signals would weight the right child as $3/5$ more likely to the left ($2/5$). Consequently the N we see at position 5 in the consensus may not really be a N but a weighted average of emissions over the nucleotides: { $C=3/5$ $A=2/5$ }.

If we consider multiple models of consensus trees (i.e. we have multiple roots per child pair) we will need to calculate a posterior odds ratio on the models to determine which one best fits the available data. This mixture of models is readily subject to analysis by evidence-confidence.

The justification for clustering similar sequences together is two-fold. Firstly, the regulatory motifs are not region specific for all genes (they do not occur in the same place in all genes or even in the same order). Secondly we might see that in addition to clustering if we incorporate localization information into the model (not absolute, rather relative relations of motifs), and then look for clusters of consensus motifs in the given regions, which we would expect localization of co-regulating motifs with respect to one another in order to facilitate protein-protein interactions.

I have illustrated a framework to explore this question in more depth. Determining the exact Bayesian framework to weigh the evidence and the variability probabilities for each nucleotide need to be further explored. However this method seems promising to identify the real regulatory motif within the variable nature of regulation.