Motif or artifact?

Martin Ph. Stehno
Department of Physics, UIUC

Physics 498BIN, Fall 2001

Abstract

Conventional motif-finding algorithms are optimized for either good soundness or completeness. Good soundness is achieved when the output lists only a few motifs that are very likely to represent binding sites. On the other hand, there are algorithms which are designed to give a complete list of possible binding sites. The downside of this is that the list will also contain typically hundreds of small variations of strong motifs, which are not considered to be motifs in their own right. Thus a real, weak motif will be found way down on the list. Here I want to report a new method of post-processing the output of an algorithm that gives a complete list. The algorithm was invented by Blanchette and Sinha[1] and clears the motif list from the artifacts of strong motifs, shrinking it to a small number of motifs that are very likely to represent the IUPAC ambiguity code[1] of binding sites.

For pedagogical reasons I chose to display most of the calculations in [1] to clarify how Bayesian statistics comes into the game, but also to point out the approximations that where used to transform this into an algorithm that can be programmed into a computer.

Artifacts and explanations

Let me pin down what an artifact is. Consider a strong motif ACGCCW, which occurs 80 times in a given sequence. Some motif-finder will report this code first place on its list of motifs. Now, let us look at a the motif ACGCCA. We would expect to find it, say, 10 times in an arbitrary sequence. Yet in the sequence we look at it will be found to be a second strong motif with an occurrence of maybe 40 times. Well, keeping in mind that ACGCCW is the ambiguity code of the actual binding site the high rate of occurrence of the sequence ACGCCA is no longer surprising - it can be 'explained'. This is the characteristic of an

[1] The IUPAC ambiguity code consists of 15 characters rather than {A, C, T, G} only, taking into account that the nucleotide sequence of a binding site can have small variations.

artifact. Here this comes from the fact that the last character is specified in the definition of site. If we could only quantify how well ‘explained’ (the occurrence of) a sequence is, we would be able to identify the artifacts and cross them out on our list of motifs leaving only the ‘best explanators’. The goal of this paper is to sketch an algorithm for the ‘best explanators problem’.

Nucleotide frequencies and scoring scheme

We will need a scoring system to measure the over-representation of a motif given the fact that other motifs are known to be part of the sequence. To get a feeling for what the task is about, it is helpful to look at a more simple problem first. Here we score according to the nucleotide frequencies only and ignore the presence other motifs[2]. The over-representation of a nucleotide sequence is the difference between the expected and the observed oligonucleotide frequency. The expected oligonucleotide frequency can be calculated using a Markov chain model. For words of length k, we can choose a subword length (or Markov order) m between 1 and $k-2$, e.g. for $k=6$ and $m=3$, we could calculate the expected frequency of GATAAC to be

$$F_{exp}(GATAAG) = \frac{F_{obs}(GATA) \times F_{obs}(ATAA) \times F_{obs}(TAAG)}{F_{obs}(ATA) \times F_{obs}(TAA)}.$$

We still have to multiply this with the number of possible word positions:

$$occ_{exp}(w) = F_{exp}(w) \times \sum_{i=1}^{S} (L_i - k + 1).$$

Here L_i is the length of the ith sequence and S is the total number of sequences.

An appropriate scoring scheme is given by the Z-score statistics, which assumes a normal distribution for the over-representation of an oligonucleotide sequence. For this problem we calculate the Z-score to be

$$Z = \frac{occ_{obs}(w) - occ_{exp}(w)}{stdev_{est}(w)},$$
$$stdev_{est}(w) = \sqrt{var_{est}(w)},$$

where w is the oligonucleotide sequence (or word), $occ_{obs}(w)$ is the observed number of occurrences of the word w, and $stdev_{est}(w)$ is an estimate for the standard deviation of occurrences of w. The estimated variance $var_{est}(w)$ is given by

$$var_{est}(w) = occ_{exp}[2K_{ov} - 1 - (2w-1) \times occ_{exp}],$$

with some self-overlap coefficients K_{ov} (see Pevzner et al. [3]). After choosing a threshold value, which will depend on the expected word size, the Z-score value can be used to measure the over-representation of an oligonucleotide sequence. It turns out that a reasonable threshold value for word size 6 is 3.4, cf. van Helden et al. (2000) [2].

Conditional probabilities and occurrences

To adapt the above scheme to our problem, we have to obtain expressions for the conditional expectation value and variance of the occurrences of a motif in a random sequence, given the occurrences of other motifs. Let the input sequence s be of length l, then E_m shall be the binary vector of length l associated with a motif m. Then annotate the occurrence of m in s by

$$\forall i = 1 \dots l, E_i^m = \begin{cases} 1, & \text{if motif } m \text{ starts at position } i \text{ in } s \\ 0, & \text{otherwise.} \end{cases}$$

The probability of some motif m to occur at position i, given a set of positions where motifs $e_1, \dots, e_\zeta$ occur, takes the form

$$\wp[E_i^m = 1 | \bigwedge_{j=1\dots l} \bigwedge_{k:E_j^{e_k}=1} E_j^{e_k} = 1] = \frac{\wp[E_i^m = 1 \wedge (\bigwedge_{j=1\dots l} \bigwedge_{k:E_j^{e_k}=1} E_j^{e_k} = 1)]}{\wp[\bigwedge_{j=1\dots l} \bigwedge_{k:E_j^{e_k}=1} E_j^{e_k} = 1]}.$$

We calculate the probabilities in the numerator and denominator starting with a sequence $T =$ NNNN...NNN. Next we write motifs $m_p (p = 1, \dots, r)$ at position i_p in T thus specializing the symbols in T. Specialization means to put in T the most general symbol that can be found at this position in both T and m_p. If no specialization is possible, the probability will be zero. Otherwise the probability is the probability of the sequence T as obtained from a Markov process. Similarly the probability p_i of a motif to be found at position i taken into account the presence and absence of the motifs $e_1, \dots, e_\zeta$ is

$$p_i = \wp[E_i^m = 1 | (\bigwedge_{\substack{j=1\dots l \\ k:E_j^{e_k}=1}} E_j^{e_k} = 1) \wedge (\bigwedge_{\substack{j=1\dots l \\ k:E_j^{e_k}=0}} E_j^{e_k} = 0)].$$

Here we can make an observation. Let a and b be two positions in the sequence. If the distance ab is larger than say a constant c_1 we have good reason to assume that there is no effect of the occurrence of motif m_a at a on a possible occurrence of motif m_b at b. The same can be inferred on absences of motifs with a constant c_2. Note that the constants c_1 and c_2 are not the same. (In fact we can reason that c_2 should be much smaller than c_1. Just remind yourself when some motif m starts at position n the probability for our motif to start at $n + 1$ will be zero, whereas if we state a 'non-occurrence' of m at n, we do not expect that this has a major influence on the probability of an occurrence of our motif at $n + 1$.) Thus p_i is

$$p_i \approx \wp[E_i^m = 1 | (\bigwedge_{\substack{j:|j-i|\leq c_1 \\ k:E_j^{e_k}=1}} E_j^{e_k} = 1) \wedge (\bigwedge_{\substack{j:|j-i|\leq c_2 \\ k:E_j^{e_k}=0}} E_j^{e_k} = 0)].$$

Making use of Bayes' theorem this becomes

$$\frac{1-\wp[\bigvee_{\substack{j:|j-i|\leq c_2\\ k:E_j^{e_k}=0}} E_j^{e_k}=1|(\bigwedge_{\substack{j:|j-i|\leq c_1\\ k:E_j^{e_k}=1}} E_j^{e_k}=1)\wedge(E_i^m=1)]}{1-\wp\,[\bigvee_{\substack{j:|j-i|\leq c_2\\ k:E_j^{e_k}=0}} E_j^{e_k}=1|(\bigwedge_{\substack{j:|j-i|\leq c_1\\ k:E_j^{e_k}=1}} E_j^{e_k}=1)\times\wp\,[E_i^m=1|\bigwedge_{\substack{j:|j-i|\leq c_1\\ k:E_j^{e_k}=1}} E_j^{e_k}=1]},$$

which can be brought into the form

$$\wp[\bigvee_{\substack{j:|j-i|\leq c_2\\ k:E_j^{e_k}=0}} E_j^{e_k}=1|C]=\sum_{X\in 2^{(j,k):|i-j|\leq c_2\wedge E_j^{e_k}=0}}(-1)^{|X|+1}\wp[\bigwedge_{(j,k)\in X}(E_j^{e_k}=1)],$$

using the inclusion-exclusion principle and C is an arbitrary condition of the usual form. Each term is computable using the specialization procedure outlined above. However, most terms will be zero since because they would lead to an illegal overlap of motifs. The actual number of overlaps of k motifs in a region $2c_2+1$ will usually be small (c_2 is typically 1 or 2). These overlaps can be generated easily. The conditional expectation of a motif m, N_m, given the occurrences of $e_1,\ldots,e_\zeta$ is

$$\mu_m=\mathbb{E}[N_m|E^{e_1},\ldots,E^{e_\zeta}]=\sum_{i=1\ldots l}p_i.$$

A similar calculation leads to the expression for the conditional variance of N_m:

$$\begin{aligned}\sigma_m^2 &=\\ &=\mathbb{E}[N_m^2|E^{e_1},\ldots,E^{e_\zeta}]-(\mathbb{E}[N_m|E^{e_1},\ldots,E^{e_\zeta}])^2=\\ &\approx\sum_{i=1\ldots l}\left(\sum_{j:|j-i|\leq c_3}\wp[E_i^m=1\wedge E_j^m=1|E^{e_1},\ldots,E^{e_k}]+\sum_{j:|j-i|\geq c_3}p_ip_j\right)-\mu_m^2.\end{aligned}$$

Finally we can write down the conditional Z-score of m, given $e_1,\ldots,e_k$,

$$Z(m|e_1,\ldots,e_k)=\frac{N_m-\mu_m}{\sigma_m}.$$

For the best explanators problem Blanchette and Sinha (2001)[1] used a greedy algorithm, which is guaranteed to give a solution, but it need not be the optimal one. (For our type and size of problem we don't have to worry about this too much.) A greedy algorithm assigns a numerical value (Z-score) to each candidate (motif), and repeats the following three steps:

1.) Check the set S, if it is a solution. (See if all the motifs from the list have been checked for being an artifact.) Otherwise, 2.) select a new candidate (pick next motif) and 3.) check, if the candidate is feasible to be in the solution (if the Z-score drops below a certain limit, conditioning in the other (remaining) motifs (on the list) when calculating the Z-score), and add it to S (keep it on the list), if so.

To reduce computer time a pre-processing step was used, removing all motifs m from the list that have a Z-score $Z(m) < Z(e)$ and $Z(m|e)$ less than some threshold. In the following 4 was used as a threshold (cf. the threshold for over-representation above.) This is supposed to filter out about 80% of the artifacts, while it is very unlikely that a real motif will be removed.

Experimental verification

Three types of experiments have been carried out to validate the algorithm. Typical inputs were 50 sequences of 800 base-pairs each with about 5 motifs planted into them. As a motif-finder the YMF algorithm of Sinha and Tompa (2000) [4] was used. It works with a restricted alphabet of 8 characters. This was shown to be sufficient for describing the binding sites of yeast.

First, an accuracy test was carried out planting 5 motifs into random sequences. In 46 out of 50 cases all the motifs were recovered correctly. A deviating motif turned out to be generalizations or specializations of the original motif and can usually be attributed to a low Z-score of the original. Meeting the expectations the accuracy decreased when fewer sequences were available. The reason for this is that the normality condition of the conditional Z-scores is no longer valid.

In a second test real biological data was used. Genes of 5 arbitrary families, whose binding sites are known, were merged into one large group of 44 genes. The algorithm produced 7 explanators, 4 of which were the known consensus for one of the gene families, the rest of the explanators on the list were part of a longer binding site in the remaining family.

Finally sets of Saccharomyces cerevisae and other yeast genes were taken to test the algorithm. Here - besides known consensus sequences - new motifs with high Z-scores have been found proofing the applicability of the algorithm.

References

[1] M. Blanchette, S. Sinha, Separating real motifs from their artifacts, Bioinformatics, 2001, Vol. 17, pp.S30-S38

[2] Van Helden et al., Statistical analysis of yeast genomic downstream sequences reveals putative polyadenylation signals, Nucleic Acids Research, 2000, Vol. 28, No. 4, pp.1000-1010

[3] P.A. Pevzner et al., J. Biomol. Struc. Dyn., 1989, 6, pp.1013-1026

[4] Sinha, S. and Tompa, M. (2000). A statistical method for finding transcription factor binding sites. In *Proceedings of the Eigth International Conference on Intelligent Systems for Molecular Biology.* AAAI Press, pp. 344-354