# Learning Weight Matrices for Identifying Regulatory Elements

Dennis F. Kibler and Steven Hampson
Information and Computer Science Department
University of California
Irvine, Ca., U.S.A.

**Abstract** *The structure of DNA regulatory patterns is partially understood, revealing an indeterminacy in the base composition. The dominant approach for representing this intrinsic variability is probability matrices, although some have used IUPAC codes and restricted regular expression languages [1]. In general the goal is to identify patterns that are distinguished from the background, where the background is usually modeled by first-order statistics. In this paper we describe and evaluate an alternative model of base variability, that of weight matrices. We also provide a novel algorithm for learning weight matrices. Unlike a probability matrix that provides a summary motif description, a weight matrix provides a motif detector. In addition, unlike the algorithms for determining probability matrices, which generally use first-order statistics to model the background, we provide an algorithm that uses kth-order statistics when learning length-k patterns. Artificial data is used to evaluate the effectiveness of this representation and training algorithm.*

*Keywords:* IUPAC, weight matrix, regulatory elements

## 1 Introduction

The particular biological problem we focus on is that of finding and characterizing regulatory elements in DNA. For lower eukaryotes these elements are contained in the Up Stream Regions (USRs) of Open Reading Frames (ORFs), and for yeast predominately in the first 500 base pairs [2]. A number of motif representation languages and approaches to motif identification have been developed to identify these patterns. However, it is difficult to make precise comparisons between them because the different approaches often produce different results given the same set of USRs and the correct biological answer is seldom known with complete precision. Consequently, in order to facilitate comparison, we generate artificial test cases where the correct answer is known and problem complexity can be incrementally adjusted along various dimensions, keeping in mind that the artificial data should be relevant to known biological constraints.

The rather successful $k$-mer oligonucleotide frequency over-represenation (OFOR) method of van Helden [3] has the advantage of modeling the background of $k$-length motifs with $k$th-order statistics. It is also fast, taking only a few seconds to exhaustively evaluate all $k$-mers ($k < 10$). Its primary disadvantage is using only exact $k$-mer sequences to represent motifs. Methods such as Consensus [4], DMS [5], Gibbs Sampling [6] and Meme [7] capture the variability of a motif in a probability matrix, but limit their representation of the background to first-order statistics with some post-editing (by hand or program) of the results. They are also comparatively slow, often requiring hours of computation to analyze a large data set.

Here we provide several extensions to the basic $k$-mer OFOR method. These methods progressively extend the representation language while maintaining $k$th-order background statis-

tics and the over-representation (OR) evaluation criterion.

## 2 $K$-mer representation

The $k$-mer OFOR approach identifies possible motif instances by looking for specific $k$-mers that are over-represented in a test set of USRs from co-regulated genes as compared to a comparison set, which is typically the set of USRs from all genes. Specifically, the observed number of occurrences of a $k$-mer in the comparison set is used to calculate the probability of seeing $X$ or more occurrences in the test set by chance, where $X$ is the observed count in the test set. If Prob($X$) is small, then the $k$-mer is over-represented in the test set. For example, OR can be defined as 1-Prob($X$) or -log(Prob($X$)). The space of possible $k$-mers can be searched exhaustively and sorted by their OR values. Because over-representation rather than coverage of the test set is computed, the approach is relatively insensitive to the presence of noise in the test-set. An added value is that the multiple occurrences of a $k$-mer in individual USRs improves the score for that $k$-mer. Repetition within a USR appears to be biologically important, although some motif-finding approaches do not use this information.

Because regulatory elements are generally effective on either strand of the USR, we can compute OR on the combined counts over both strands, or equivalently, the combined counts of a $k$-mer and its reverse complement. In this context, it is apparent that OR can be computed for any set of $k$-mers rather than just individual $k$-mers. In particular, this is true for any motif-classification language where there is a clear-cut answer to whether a $k$-mer belongs to the motif or not.

## 3 Prototype Representation

A useful set of instances for which OR can be easily and quickly computed is prototype representation. The general idea is that there is some ideal, prototypic sequence, but that a small amount of random variation is permitted. For example, one or possibly two mismatches might still be functional while a greater number would disrupt binding too much. We define the $M_n$ shell around a $k$-mer to be the set of $k$-mers that differ from the central, prototypic $k$-mer in exactly $n$ places. Hence the $M_0$ shell is the string itself, the $M_1$ shell is the set of $k$-mers that have exactly one mismatch with the string, the $M_2$ shell is the set of $k$-mers with exactly two mismatches, etc. The counts corresponding to each $M_n$ shell are referred to as $C_n$.

The $C_1$ value for a given $k$-mer can be computed by summing the $C_0$ values of its $M_1$ neighbors. In general, $C_n$ can be computed by summing the $C_0$ values of the $\begin{pmatrix} k \\ n \end{pmatrix} * 3^n$ $M_n$ neighbors. However, by judicious use of lower-order $C_n$ values, any $C_n$ value can be computed by summing over just the $M_1$ neighbors, which significantly improves performance. For example, for $k = 10$ and $n = 3$, there are 3240 $M_3$ neighbors but only 30 $M_1$ neighbors.

The OR measure can be applied to $C_1$ just as it was to the exact-match $C_0$ count. Because of the possible variation in a motif, if an over-represented $M_0$ motif instance is found, its $M_1$ shell is apt to be over-represented as well. This is especially true for the most central or prototypic instances of the motif. For motifs with a large amount of variability, the $M_1$ shell can be more over-represented than the $M_0$ center or any of the individual $k$-mers that make up the $M_1$ shell. This is especially important with longer motifs. OR of the $M_1$ shell provides an independent measure of the importance of the $M_0$ instance. If both $M_0$ and $M_1$ are highly over-represented, the chances of $M_0$ being a real motif instance are increased. Alternatively, the $M_0$ and $M_1$ patterns can be combined and a single OR score computed. Whether this is useful or not depends on the degree of variability of the motif. If the motif has low variability, the OR of the $M_1$ shell will be correspondingly low.

## 4 IUPAC Representation

The $M_n$ shells allow for a certain amount of variation around the central $k$-mer prototype. This effectively deals with random variation, but does not directly address the issue of degenerate motifs where some positions may vary in specific ways and other positions may not vary at all. An IUPAC motif description explicitly allows for arbitrary disjunctions at each position of the motif. This allows for degenerate motifs, but does not directly address the issue of random variation. With IUPAC representation there is no protopypic $k$-mer; all positive $k$-mers are equally peripheral.

As with the $M_n$ prototypic shells, an IUPAC motif defines a set of $k$-mers for which group OR can be computed. However, unlike exact or prototypic $k$-mer representation in which all $4^k$ $k$-mers can be exhaustively evaluated, there are $15^k$ length-$k$ IUPAC motif descriptions, which cannot be exhaustively evaluated for anything but very small $k$. Consequently a hill-climbing approach is used: given a plausible $k$-mer motif instance as a starting point, IUPAC hill-climbing sequentially tries all possible IUPAC codes at each position of the motif, chooses the code with the biggest OR value, and repeats until no changes are possible.

## 5 Weight Matrix Representation

Like a probability table, a weight matrix uses a 4 x $k$ table of real numbers to represent a length-$k$ motif. However, unlike the probability table that summarizes the frequency of each base at each position over a set of known motif instances, a weight indicates the importance of each base at each position. A DNA sequence is classified by summing the corresponding weights for each base in the sequence, and if the sum is greater than or equal to a threshold the sequence is considered a positive example of the motif.

In this context, an exact $k$-mer can be realized as weight matrix with three 0s and a 1 in each column and a threshold of $k$. An $M_1$ prototype (including the $M_0$ center) has a threshold of $k-1$ and an $M_2$ prototype has a threshold of $k-2$. An IUPAC motif is equivalent to a weight matrix with weights of 0 or 1 and a threshold of $k$, but having any number of 1s in a column. A prototypic IUPAC motif could allow for a certain amount of random variation by setting the threshold to $k-1$ or $k-2$. The final generalization to an unrestricted weight matrix allows the different bases within a column to be weighted differently.

A weight matrix can be viewed as a model of the binding site of a regulatory protein. Each base in a candidate motif instance makes some positive, negative or neutral contribution to the binding stability of the DNA-protein complex. The weights in the weight matrix can be thought of as modeling those effects. If the sum of the individual contributions is greater than a threshold, the DNA-protein complex can be considered stable enough to be functional. More likely, functionality would vary continuously (although not necessarily linearly) with the stability of the DNA-protein complex, but for computational purposes, a binary classification is a convenient simplification.

In this context, the possibility of negative weights is not unreasonable; a particular base at a specific position may destabilize binding. Negative weights will obviously never occur if a probability table, but from a formal point of view, the possibility of negative weights does not in itself expand the representation power since a weight matrix (plus threshold) with negative weights can always be converted to an equivalent one without negative weights. More generally, it is possible to convert any weight matrix to an equivalent one that conforms to the constraints of a probability matrix (all entries positive, columns sum to 1.0).

A weight matrix can also be viewed as defining a hyperplane that classifies $k$-mer patterns as positive or negative. The basic $k$-mer OR approach implicitly uses a hyperplane to consider each individual $k$-mer pattern as the positive set. Prototype and IUPAC representations relax the restrictions on the weights of

the hyperplane. The final generalization is to allow arbitrary hyperplanes. In all cases the goal is to find the hyperplane whose positive set is most over-represented.

Since all possible hyperplanes cannot be exhaustively tested to find the optimum, hill-climbing is used. The assumption is that a hyperplane can be significantly improved with a series of small adjustments that monotonically improve its OR measure. Specifically, given an initial promising $k$-mer, prototype or IUPAC motif, the corresponding weight matrix/hyperplane is generated and the nearest $m$ points ($k$-mer patterns) are considered as positive and negative training instances. If the OR measure is improved by adjusting the hyperplane to reclassify a point, the change is made and the process continues using the resulting new set of neighboring points. A reasonable value for $m$ is around 40, with larger values only slightly increasing the hill-climbing power. Various methods of adjusting the hyperplane to reclassify a point were investigated and the single best method was to try all single-base adjustments to include/exclude each of the points. In general, weight matrix hill-climbing can be viewed as incrementally adjusting a (linear) hyperplane so as to maximize an unknown non-linear function based on its classification of the points, in this case the resulting OR measure.

The perceptron training algorithm is a provably convergent method of training a hyperplane given a set of points that can be correctly classified by a hyperplane. This provides a method of creating a weight matrix given a set of known motif instances and non-instances. However, unlike the perceptron training algorithm in which the correct classification of each point is known, with OR hill-climbing it is entirely possible that including a point will improve the measure, then later, excluding the same point will improve the measure again since the measure depends on the classification of all other points, which changes on each adjustment. Weight matrix hill-climbing works well in the sense that it climbs to very high OR measures, but tends to find slightly different motifs when the search order or starting points are varied slightly, expecially when the correct motif is highly degenerate.

Entropy and Expectation Maximization (EM) have both been used as the objective function for optimizing probability matrices, and it has been shown that the two are closely related. A relationship can also be shown between probability and weight matrices: Specifically, a $k$-length weight matrix defines a set of $k$-mers that it classifies as positive instances, and the probability matrix for that set of $k$-mers is a close approximation of the original weight matrix [9]. Consequently, a probability matrix optimized for entropy or EM may also reasonably function as a weight matrix. However, the correlation decreases as the distribution of observed positive $k$-mer becomes less uniform. That is, if all positive $k$-mers occur with equal frequency, the correlation is very good, although not necessarily good enough for perfect classification. However, if some positive $k$-mers are over or under-represented in the biological data, the probability matrix changes while the weight matrix does not. Thus if the ultimate goal is classification of motif instances, it seems reasonable to directly optimize the matrix for that property rather than for entropy or EM.

## 6 $K$-mer Overlap

The issue of overlapping binding sites is important for both theoretical and practical reasons. A simple example serves to illustrate the problem: If a given region of DNA contains two overlapping binding sites for the same transcription factor, should this be considered as one or two sites? This has theoretical implications for the optimum structure of both the binding region and the transcription factor. It has been observed that multiple independent copies of a motif have a cumulative effect [10], so it is not unreasonable that multiple overlapping binding sites could be more effective than a single site, at least in a probabilistic binding model. Likewise, if a transcription factor could

bind the same sequence in multiple ways, this could increase the probability of interaction.

The practical issue is that treating all binding sites as equal and independent, even if they overlap, greatly simplifies computation. For example, one method of determining how many times an IUPAC motif matches a given data set is to slide the motif along the DNA, counting matches. It is easy to detect overlapping matches in this case, but scanning the complete data base is time consuming if done repeatedly. Alternatively, the data base can be scanned once and the frequency of all $k$-mers counted. The number of matches of the IUPAC motif can then be calculated by summing the counts of the $k$-mers it matches. This can be done very quickly, but the matches may overlap.

Empirically, the two approaches can give different answers to the optimum IUPAC or weight matrix motifs to maximize OR, but often they give the same answer. Thus, the fast overlapping method might be used as an approximation of the slower, non-overlapping method, even if the non-overlapping model ultimately proves to be more biologically accurate. One particularly productive application is do non-overlapping hill-climbing, but to prefilter each potential change using an overlapping evaluation. Only changes that lead to an improved overlapping score are considered for non-overlapping evaluation. This speeds up the hill-climbing process by at least 90% and does not appear to significantly effect the final results. This can be done for both IUPAC and weight matrix hill-climbing.

# 7 Evaluation on Synthetic Data

Our goal is to demonstrate the effectiveness of IUPAC and weight matrix OR learning using synthetic data that is similar to real biological data, but has the advantage that we know the correct answer.

Experiments are done using a test of 40 artificial 500-base USRs and a comparison set, containing the test set, of 1000 USRs. USRs are artificially generated, but real USRs could also be used. A hidden motif instance is added to each of the test USRs. Motif instances are generated from an IUPAC motif description plus a controlled amount of random variation. Both the degree of degeneracy of the IUPAC motif and the amount of random variation are adjusted and studied independently and in combination. Here we only report experiments where we vary the degree of IUPAC indeterminancy. These IUPAC motifs should, in principle, be detectable by most motif-finding algorithms.

Two main questions need to be addressed:

1. Is over-representation a reasonable objective measure to optimize IUPAC and weight-matrix motifs?

2. Can the optimum motif be found?

   (a) Can any motif instances (exact $k$-mers) be identified?

   (b) Can hill-climbing from a motif instance recover the complete motif?

We will describe, summarize and discuss one synthetic experiment here that addresses all of these issues. An application of IUPAC hill-climbing to biological data is found in [8].

In this experiment an 8-mer motif instance with an increasing amount of IUPAC degeneracy was added to the test set. Specifically, the 8-mer atgccgta was incrementally generalized to allow a single disjunct at 6 of the 8 positions, resulting in the IUPAC motif a[tg][ga][ct][ca][gc][tg]a. A first-order uniform background (a = t = g = c = .25) was used, and no random variation outside of the IUPAC motif was permitted. In each case the task was to recover the correct IUPAC motif starting with a single motif instance. There are $15^8$ possible IUPAC formula, so it is impossible to search the space exaustively.

The results of this experiment are summarized in Table 1. In this table, the $C_0$ values for test and comparison sets plus the resulting

probability value are shown for the highest-scoring 8-mer when sorted on OR. The same three values are shown for the IUPAC motif that results from hill-climbing from that $k$-mer. This is repeated with 0 to 6 disjuncts.

The first row shows the results with 0 disjuncts. Because the artificial USRs are generated with a uniform first-order model, the test set $C_0$ count alone is a reasonable predictor of OR. With a different background model this might not be the case. With this background model, the test set without the motif contains two 8-mers that occur 5 times and 21 that occur 4 times. The average 8-mer is expected to occur about .3 times $= 40 * 500 * 1/(4^8)$. Thus any 8-mer that occurs more than 5 times is statistically unusual and will be identified. With 40 occurrences, the motif is easy to identify. The $k$-mer occurs 5 times in the comparison set (expected = 7.6) outside of the 40 occurrences due to the test set. The fact that the IUPAC values are the same as the $k$-mer values shows that IUPAC hill-climbing could not generalize the 8-mer to yield a better OR value. This is as expected and contrasts with the results if hill-climbing was started at a shifted version of the motif such as tgccgtaa. In this case IUPAC hill-climbing correctly generalized the instance to tgccgta[atgc].

In the second row, a single disjunct is added. As expected, this divides the 40 motif instances among two approximately equal $k$-mers ($C_0 =$ 22 and 18). Again, these values are very high compared to the background and so are easily identified. Like the maximum $C_0$ value, the maximum OR value degrades but is still highly significant. More importantly, the correct IUPAC motif is reliably recovered by hill-climbing. The number of IUPAC matches in the test set is 40, so it perfectly identifies the complete set of motif instances. The IUPAC OR value degrades slightly because the additional IUPAC motif member matches seven additional 8-mers in the comparison set.

A similar trend continues through four disjunctions. Both the maximum $C_0$ and OR for motif instances decline as the number of disjuncts, and hence the number of different motif instances, increases. In addition, the OR of the correct/optimum IUPAC motif continues to decline. It is impossible to exhaustively verify that the correct IUPAC motif has the greatest OR over all IUPAC formula, but we strongly suspect this is the case. With four disjuncts, the $C_0$ and OR of the most frequent motif instance is right at the boundary of detectability.

With five disjuncts, individual motif instances are no longer sufficiently overrepresented to be distinguished from the random background. Consequently, the top $k$-mer is not a motif instance. This can be addressed by sorting on the combined $M_0$ and $M_1$ shells for each $k$-mer. With this modification, the top $k$-mer is a motif instance. Starting with this instance, IUPAC hill-climbing usually climbs to the expected answer but occasionally gets stuck at a noticeably worse local maxima.

In the final row with six disjuncts, even combined $C_0$ and $C_1$ can't identify motif instances, and starting from known motif instances usually gets stuck at local maxima. However, it occasionally climbs to the expected IUPAC answer. No better IUPAC motifs were observed, so this may well the the global maximum. Since IUPAC hill-climbing is fast, it is not actually necessary that the top $k$-mer is a motif instance or that hill-climbing from a motif instance always lead to the correct IUPAC motif. It is feasable to test the top 50 or so $k$-mers with five or ten restarts each to see which leads to the best IUPAC motif. Using overlapping counts, this can be done in about a minute. In that sense, the problem can be solved with six disjuncts.

Despite a different motif representation language and hill-climbing algorithm, weight matrix results largely paralled IUPAC results. With 0 to 3 disjuncts, the final classification of the weight vector was identical to the IUPAC answer. With four disjuncts, the weight matrix achieved a slightly better OR score (1.5e-25 vs 1.8e-25) by covering one extra $k$-mer in the test set. With five disjuncts, weight matrix hill-climbing frequently gets stuck at poor local maxima, but occasionally climbs to values that

Table 1: Best $k$-mer and IUPAC values with increasing number of disjuncts. D = number of disjuncts, TC = test set $C_0$, CC = comparison set $C_0$, Prob = Prob(TC).

| top $k$-mer | | | | hill-climb IUPAC | | |
|---|---|---|---|---|---|---|
| D | TC | CC | Prob | TC | CC | Prob |
| 0 | 40 | 45 | 3.3e-39 | 40 | 45 | 3.3e-39 |
| 1 | 22 | 23 | 7.6e-21 | 40 | 52 | 8.2e-37 |
| 2 | 11 | 14 | 2.5e-11 | 40 | 59 | 9.7e-35 |
| 3 | 9 | 17 | 4.6e-08 | 43 | 94 | 2.1e-30 |
| 4 | 6 | 12 | 1.1e-05 | 46 | 148 | 1.8e-25 |
| 5 | 3 | 9 | 5.9e-03 | 48 | 256 | 1.1e-17 |
| 6 | | | | 58 | 496 | 2.7e-12 |

are similar to (and sometimes slightly better than) the IUPAC answer. With six disjuncts, weight matrix hill-climbing always terminated at a poor local maxima.

## 8 Conclusions

A number of general conclusions can be drawn from these results. First, for this series of problems, maximum IUPAC and weight matrix OR corresponds to the known correct answer. Thus, from this and other experiments it appears that OR is a reasonable objective function to optimize for motif classifiers. Second, as expected, motif instances become increasingly hard to detect as motif degeneracy increases. Including $C_1$ values extends the reach of exact $k$-mer identification. Third, the IUPAC and weight matrix OR hill-climbing algorithms are reasonably effective since, given a single motif instance, they can generally recover the correct motif. The risk of being trapped at a poor local extrema increases as the maximum/optimum OR decreases. This can be mitigated by using multiple restarts.

## References


[1] Brazma, A., Jonassen, I., Vilo, J. & Ukkonen, E. Predicting gene regulatory elements in silico on a genomic scale. Genome Research, 1202-1215, 1998.

[2] Latchman, D. *Eukaryotic Transcription Factors.* Academic Press, 1999.

[3] van Helden, J., Andre, B. & Collado-Vides, J. Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. JMB. 281:827-842, 1998.

[4] Stormo, G. D. & Hartzell, G. W. Identifying protein-binding sites from unaligned DNA fragments. Proc. Nat. Acad. Sci. 86, 1183-1187, 1989.

[5] Hu, Yuh-Jyh, Kibler, D & Sandmeyer, S. Detecting Motif from Sequences. ICML. 181-190, 1999.

[6] Lawrence, C., Altschul, S., Boguski, M., Liu, J., Neuwald, A. & Wooton, J. Detecting Subtle Sequence Signals: A Gibbs Sampling Strategy for Multiple Alignments. SCIENCE. 262: 208-214. 1993.

[7] Bailey, T.L. & Elkan, C. Unsupervised Learning of Multiple Motifs in Biopolymers Using Expectation Maximization. Machine Learning 21, 51-80, 1995.

[8] Hampson, S., Baldi, P., Kibler, D. & Sandmeyer, S. Analysis of Yeast's ORF Upstream Regions by Parallel Processing, Microarrays, and Computational Methods. ISMB, 190-201, 2000.

[9] Hampson, S., & Volper, D. Linear Function Neurons: Structure and Training. Biol. Cyber. 53, 203-217, 1986.

[10] Hughes, J., Estep, P. W., Tavazoie, S. & Church, G. M. Computational Identification of Cis-regulatory Elements Associated with Groups of Functionally Related Genes in Saccharomyces cerevisiae. JMB 296 1205-1214, 2000.