World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:3, No:4, 2009

Exploring the combinatorics of motif alignments for accurately computing E-values from p-values

T. Kjosmoen, T. Ryen, and T. Eftestøl

Abstract—In biological and biomedical research motif finding tools are important in locating regulatory elements in DNA sequences. There are many such motif finding tools available, which often yield position weight matrices and significance indicators. These indicators, p-values and E-values, describe the likelihood that a motif alignment is generated by the background process, and the expected number of occurrences of the motif in the data set, respectively. The various tools often estimate these indicators differently, making them not directly comparable. One approach for comparing motifs from different tools, is computing the E-value as the product of the p-value and the number of possible alignments in the data set. In this paper we explore the combinatorics of the motif alignment models OOPS, ZOOPS, and ANR, and propose a generic algorithm for computing the number of possible combinations accurately. We also show that using the wrong alignment model can give E-values that significantly diverge from their true values.

Keywords—Motif alignment, combinatorics, p-value, E-value, OOPS, ZOOPS, ANR.

I. Introduction

There are many different algorithms available for finding motifs in gene sets [1], [2]. Often, the comparison of the different algorithms entails running each algorithm on an artificially constructed data set where the motifs to be found are known in advance. In these cases, comparing the methods can be done by measuring how well the detected motifs match the inserted motifs.

If the motifs present in the data set are *not* known in advance, alternative methods have to be used in order to compare the algorithms. The first obstacle in this endeavor is the fact that these tools, such as MEME [3], Weeder [4], NestedMICA [2], etc., all produce different outputs. When evaluating a motif, there are mainly two properties that are the most important: The *p-value* and the *E-value*. The p-value signifies the probability that the given motif could have been created by the random background model, while the E-value is the number of times one should expect the given motif to appear in the data set, if the data set has been randomly generated.

Most, if not all, motif finding algorithms represent the motif as some sort of position weight matrix (PWM) [5], be it a position count matrix (PCM) [3], position frequency matrix (PFM) [2], or a PWM denoting the information content [3]. In addition to some form of PCM or PFM, the motif finding tools can include one or more p-values and/or E-values. If the p-values and E-values *are* included in the output, they are not always directly comparable. The reason for this is that there are many different ways of estimating both p-values and E-values [6], [7], and these methods do not all compute the exact same estimate values.

T. Kjosmoen, T. Ryen, T. Eftestøl are with the Department of Electrical Engineering and Computer Science at the University of Stavanger, 4036 Stavanger, Norway. E-mail: thomas.kjosmoen@uis.no

II. Motif alignments

The PWM represents the resulting motif in terms of a matrix in which one dimension is the letter Σ_i in the given alphabet $\Sigma = \{A, C, G, T\}$, and the other dimension is the position j in the the motif. Each element in the matrix represents the contribution the letter Σ_i has on the motif in the position j. Consider a set of N_m sub-sequences of length L_m found in the given gene set that are sufficiently similar to each other that one can assume they are all representations of a single motif. These sub-sequences, or rather their positions and extents in the gene set, will be referred to as *match sites* or simply just *sites* throughout this article. As an example, consider the 7 sub-sequences shown in Table I.

TABLE I
Sub-sequences making up the example motif.

Site	Motif position j					
k	1	2	3	4	5	6
1	G	G	C	C	A	A
2	G	G	T	C	A	A
3	G	G	A	C	A	A
4	G	A	C	C	A	A
5	G	A	T	C	A	A
6	G	G	T	G	A	A
7	G	A	A	C	A	A

When performing the motif alignment of the example sub-sequences in Table I, the position count matrix is a count of all occurrences of the letter Σ_i at position j in the alignment. The PCM for the example alignment is shown in Table II.

TABLE II
Position count matrix for motif described by Table I.

Symbol	Motif position j					
Σ_i	1	2	3	4	5	6
A	0	3	2	0	7	7
C	0	0	2	6	0	0
G	7	4	0	1	0	0
T	0	0	3	0	0	0

A position frequency matrix contains the frequencies $p_{i,j}$ in which each letter occurs at position j in the motif alignment.

The frequencies can be computed by using $p_{i,j} = n_{i,j}/N_m$. Finally, the elements in the position weight matrix (or position-specific weight matrix) represent some form of measure, or score, of the weighted distance between the motif alignment and the background distribution. There are several measures of the position-specific scores for each symbol that are used, such as χ^2, log-likelihood, Kullback-Leibler divergence, euclidian distance, etc.[5]

In order to make an even comparison of motifs found by different algorithms, either the PCM or PFM can be used to compute the p-values, for instance by employing one of the previously mentioned scores such as the Kullback-Leibler.

While knowing the probability of a motif having occurred by chance is an important measure, it does not allow for a direct comparisons between motifs of different lengths. A more useful measure is thus the E-value; how many times the motif is expected to occur in a randomly generated gene set. If the p-value is available, the E-value can be computed by multiplying the p-value by the total number of possible alignments. The E-value can thus be defined by using (1).

$$e\text{-}value = p\text{-}value \cdot C_p \tag{1}$$

The number of possible alignments of a motif, C_p, depends on several factors. The first factor is the number of sites N_m in the motif, as well as the length L_m of the motif. Another important factor is the mode of which the sites are chosen, i.e. whether one site per sequence or multiple sites per sequence is allowed. As an example, the motif finding algorithm MEME has three basic modes of operation: One occurrence per sequence (OOPS), zero or one occurrence per sequence (ZOOPS), and any number of repetitions (ANR). The next few sections will cover these three alignment modes and how the number of possible alignments, C_p, for each one can be computed.

A. One occurrence per sequence

The alignment mode of OOPS is quite straight forward: There shall be only one match site per gene sequence, no more, no less. While the number of combinations in this mode seems obvious, it will serve as a lead-up to the more complicated cases of ZOOPS and ANR.

Consider a gene sequence of length L_s empty of any match sites. A simple way of representing this is as a series of dots; L_s dots, to be specific. The gene sequence is illustrated in Fig. 1. When computing the number of ways in which a single

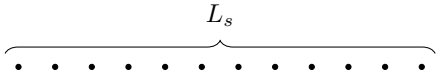

Fig. 1. Available positions in a given gene sequence.

match site of length L_m can be placed in a gene sequence of length L_s, the intuitive way may be to just count the number of positions from the beginning of the gene sequence and up until the end is reached. Calculating this number is as simple as $L_s - L_m + 1$. There is another way of looking at this problem, which will also make it easier to understand the combinatorics of the cases where there can be multiple match sites in a single sequence.

To illustrate, consider a match site placed on a gene sequence as shown in Fig. 2. The site covers L_m nucleotides, or letters Σ_i, leaving $L_s - L_m$ nucleotides open. Because

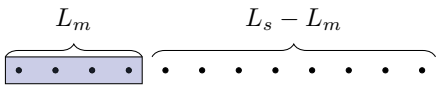

Fig. 2. A single motif site placed on the gene sequence.

the match site will *always* cover L_m consecutive nucleotides, one can instead think of the match site as a single entity. In other words, conceptually remap the L_m nucleotides into a new single nucleotide, as shown in Fig. 3. Clearly, this is a

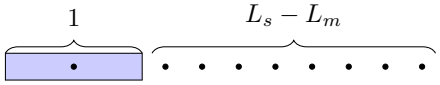

Fig. 3. Remapped nucleotide positions with a single motif site.

new construct; the length of the new conceptual gene sequence has altered. There are now $L_s - L_m + 1$ nucleotides, which actually is the same number was reached earlier. Thinking of this in combinatorics terms, C_p can be computed as the the number of combinations in which 1 site can be chosen out of $L_s - L_m + 1$:

$$C_{p,single} = L_s - L_m + 1 = \binom{L_s - L_m + 1}{1} \tag{2}$$

The purpose of including the binomial in the right-most part of (2) is to illustrate the ties between OOPS and ZOOPS, and will be revisited later in the article.

An OOPS alignment almost always involves multiple gene sequences. An illustration of an OOPS alignment is shown in Fig. 4. For each position of a match site in the first sequence,

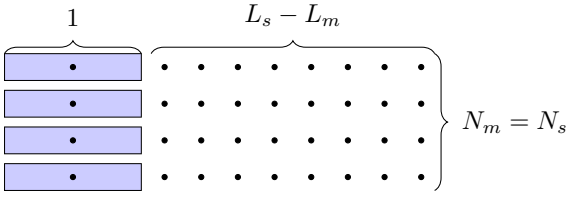

Fig. 4. Gene sequence set with a remapped single motif site per sequence.

there are many combinations of which the rest of the sites can be positioned in the remaining sequences. In fact, the total number of combinations in the whole gene set is the product of all the combinations for each individual sequence, as given by:

$$C_{p,OOPS} = (L_s - L_m + 1)^{N_m} = \binom{L_s - L_m + 1}{1}^{N_m}. \tag{3}$$

Note that in the case of OOPS, the number of sequences N_s is equal to the number of match sites N_m, but to be consistent with ZOOPS and ANR, N_m has been chosen as the exponent here.

B. Zero or one occurrence per sequence

The case of ZOOPS is quite similar to the OOPS mode. In fact, OOPS can be seen as a special case of ZOOPS. As the name implies, there can now be either zero or one match site per gene sequence. An example of a ZOOPS alignment is shown in Fig. 5. Obviously, there cannot be more site matches

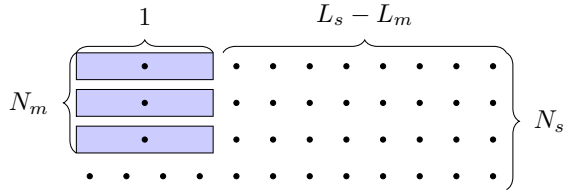

Fig. 5. Gene sequence set with one or zero remapped motif sites per sequence.

than sequences, and an alignment must per definition always contain at least two match sites in order to be an alignment, so the following constraint applies:

$$2 \leq N_m \leq N_s. \tag{4}$$

There is now another dimension to the problem of computing the number of alignment combinations. Looking at the example in Fig. 5 it is clear that the number of ways the N_s sequences can be picked must be taken into account.When aligning the sites, one must choose N_m out of the N_s available sequences. For each of these sequence combinations, the number of combinations of the match sites is still the same as with OOPS shown in (3). The total number of combinations for a ZOOPS alignment is thus the number of ways the sequences can be picked multiplied with the number of combinations for each of those picks, resulting in (5) as well as the simplification (6).

$$C_{p,ZOOPS} = \binom{N_s}{N_m} \binom{L_s - L_m + 1}{1}^{N_m} \tag{5}$$

$$= \binom{N_s}{N_m} (L_s - L_m + 1)^{N_m} \tag{6}$$

It is now evident that when using the OOPS alignment mode, when N_s equals N_m, the first binomial part of (5) becomes 1, and we are left with (3).

C. Any number of repetitions

While the cases of OOPS and ZOOPS were rather trivial, the fact that the ANR alignment mode can have multiple site matches in a single sequence makes it significantly more complex. The alignment of two match sites in a single gene sequence is illustrated in Fig. 6. Fig. 6 shows that the sites

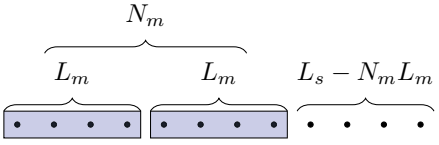

Fig. 6. Multiple motif sites placed on the gene sequence.

cover an area of N_m times L_m, leaving $L_s - N_m L_m$ open nucleotides. The observant reader may notice that if a site is placed less than L_m nucleotides away from either of the ends, several available positions for the remaining sites are lost. Again the same trick of conceptually remapping the positions can be used by assuming that each match site occupies only a single nucleotide, as illustrated in Fig. 7. By essentially counting the open positions, the problem of the lost border alignments is successfully avoided.

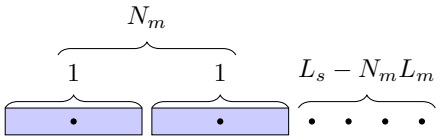

Fig. 7. Remapped nucleotide positions with multiple motif sites.

It is clear from Fig. 7 that the problem has again reduced to a simple matter of computing a binomial coefficient. In other words, choosing N_m sites out of a total of $L_s - N_m L_m + N_m$ positions, as shown in (7).

$$C_{p,many} = \binom{L_s - N_m L_m + N_m}{N_m} \tag{7}$$

Now that it has been shown how to count the number of ways in which N_m sites can be chosen from a single gene sequence, the issue of choosing the sequences arises. Consider the case where the sites are chosen from multiple, or all, of the sequences in the data set. If one selects n_1 sites from each of a_1 sequences, n_2 from a_2 other sequences, and so on, the selections must satisfy the following constraint:

$$a_1 \cdot n_1 + a_2 \cdot n_2 + \ldots + a_k \cdot n_k = N_m \tag{8}$$

What (8) really means, is that no more than the available N_m sites can be picked.

There are also two other constraints that need to be satisfied. The maximum number of gene sequences to select from is already given by N_s, which means that the following constraint must be satisfied:

$$a_1 + a_2 + \cdots + a_k = N_s \tag{9}$$

There is also a limit to the number of sites that will fit inside a single gene sequence, given by the following equation:

$$N_{max} = \left\lfloor \frac{L_s}{L_m} \right\rfloor \tag{10}$$

As there cannot be more than N_{max} sites in a single sequence, nor more than the N_m to be aligned, the last restraint is then the maximum value for the n_j's given by (11).

$$n_{max} = \min N_{max}, N_m \tag{11}$$

Because counting 0 sites in a sequence does not contribute to the total number of combinations, only the n_j's from 1 up to n_{max} need to be evaluated. This means that the value of k in (8) and (9) is thus equal to n_{max}.

Consider an example of an ANR alignment, illustrated in Fig. 8. Please note that the sequences in Fig. 8, once each

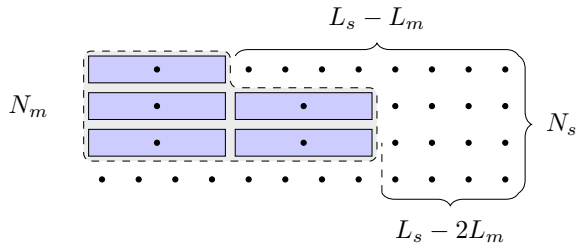

Fig. 8. Gene sequence set with an ANR alignment, with remapped match sites.

site length has been remapped to occupy a single nucleotide position, are actually of different lengths.

As the example in Fig. 8 illustrates, there is one sequence with a single site and two sequences with two sites each. Using the format of the constraint in (8), this can be expressed as $2 \cdot 2 + 1 \cdot 1 + 1 \cdot 0 = 5$, where $a_1 = 2$, $n_1 = 2$, $a_2 = 1$, $n_2 = 1$, $a_3 = 1$, and $n_3 = 0$.

Assuming the partitioning of N_m as shown in (8), the total number of ways that such a set of site partitions could be combined can be calculated.

Again considering the ANR example shown in Fig. 8, the number of ways the sequences are chosen must be calculated. This can be done by first find the number of ways to choose which sequences will have n_1 sites, which sequences will have n_2 sites, and so on. The way to do this is to consider these sequences as separate entities, and thus instead look at the number of these entities, the a_j's. Using the constraint in (8) one must first pick a_1 of the gene sequences in the data set, then a_2 of the remaining N_{rem} sequences, and so on until $a_{n_{max}}$ has been reached. The number of ways these sequences can be picked from the gene set, $C_{p,seq}$, can be found by computing the partial binomial coefficients $\binom{N_{rem}}{a_j}$ and multiplying them, as given by (14). Using the chosen example, the sequences can be picked in $\binom{4}{2}\binom{2}{1}\binom{1}{1} = 12$ different ways. Again, strictly speaking, only the non-zero a_j's need to be picked, as the sequences in which no sites will be aligned do not contribute to the overall number of combinations.

For each of the a_j sequences in which n_j sites are to be aligned, the number of possible alignments, $C_{p,many}$, is given by (7). Again, as with $C_{p,ZOOPS}$ shown in (5), the binomial coefficients $\binom{N_{rem}}{a_j}$ must be multiplied with the number of combinations for each sequence. To clarify, for each j,

$$\binom{N_{rem}}{a_j}\binom{L_s - n_j L_m + n_j}{n_j}^{a_j}. \tag{12}$$

has to be computed. Inserting (12) into (14) gives the total number of combinations for a single partition set, $C_{p,part}$, as shown in (15).

Now that the method for computing the number of ways in which a single partition of sites can be chosen from the data set has been established, the matter of integer partitioning must be considered. Partitioning an integer into addends [8] [9], such as the one shown in (8), is an old problem in number theory and has been studied by great mathematicians such as Euler, S. Ramanujan, J.H. Hardy, J. E. Littlewood, and H. Rademacher.

One way to think about the integer partitioning problem, is to consider all the different ways in which 5 marbles can be placed into different boxes. For instance, all the marbles can be put into a single box, or 4 in one and 1 in another, etc. All the different ways in which the number 5 can be partitioned is illustrated in Fig. 9 (a) with the numerical equivalents listed

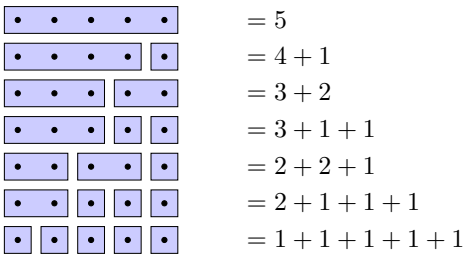

Fig. 9. Integer partitioning of the number 5.

alongside in Fig. 9 (b).

A more structured way of displaying the set of partitions of N_m, is to use a matrix, $\boldsymbol{M_{part}}$, with the n_j's as one dimension, and the partition number l as the second dimension. The partition number runs from 1 up to p_{N_m}, where p_{N_m} is the number of possible partitions of N_m given the constraints of eqs. (9) and (11). Being a well studied problem [8] [9], the computation of p_{N_m} will not be covered in this paper. Each element in $\boldsymbol{M_{part}}$ is the a_j's from the definition given in (8), and each row represents the a single partitioning.

TABLE III
PARTITIONING MATRIX $\boldsymbol{M_{part}}$ FOR N_m AS GIVEN BY (8) AND CONSTRAINED BY (9) AND (11).

Partition	Sites per sequence, n_j			
number, l	1	2	...	n_{max}
1	$a_{1,1}$	$a_{1,2}$	...	$a_{1,n_{max}}$
2	$a_{2,1}$	$a_{2,2}$	...	$a_{2,n_{max}}$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
p_{N_m}	$a_{p_{N_m},1}$	$a_{p_{N_m},2}$	...	$a_{p_{N_m},n_{max}}$

To help illustrate Table III better, consider the example from Fig. 8 once again. Taking the digits, or addends, from each partitioning shown in Fig. 9 and showing them in matrix form, yields the $\boldsymbol{M_{part}}$ matrix shown in Table IV.

There are many different algorithms available for computing all the partitioning sets, for instance the fast ZS1 and ZS2 algorithms by Zoghbi and Stojmenovic [10].

Once the matrix $\boldsymbol{M_{part}}$ of all valid partitions has been constructed, computing the total number of combinations for a given set of N_s, L_s, N_m, and L_s is a simple matter of adding up the contributions from each partition set, as shown in (13).

$$C_{p,ANR} = \sum_{l=1}^{p_{N_m}} C_{p,l,part} \tag{13}$$

The astute reader may have noticed that the formula presented in (13) is not the most effective way to compute this

$$C_{p,l,seq} = \binom{N_s}{a_1}\binom{N_s - a_1}{a_2}\binom{N_s - a_1 - a_2}{a_3}\cdots\binom{N_s - \sum_{i=1}^{n_{max}-1} a_i}{a_{n_{max}}} = \binom{N_s}{a_1} \cdot \prod_{j=1}^{n_{max}} \binom{N_s - \sum_{i=1}^{j-1} a_i}{a_j} \quad (14)$$

$$C_{p,l,part} = \binom{N_s}{a_1}\binom{Ls - n_1 L_m + n_1}{n_1}^{a_1} \cdot \prod_{j=1}^{n_{max}} \left[\binom{N_s - \sum_{i=1}^{j-1} a_i}{a_j}\binom{L_s - n_j L_m + n_j}{n_j}^{a_j}\right] \quad (15)$$

Algorithm 1 Algorithm for computing the precise number of possible ANR alignment combinations of a motif.

procedure ALIGNMENTCOMBINATIONSANR($\boldsymbol{M_{part}}$,N_s,L_s,N_m,L_m)
  $C_{p,ANR} \leftarrow 0$ ▷ Initialize total number of combinations
  $n_{max} \leftarrow \min\left\lfloor \frac{L_s}{L_m} \right\rfloor, N_m$ ▷ Compute the maximum number of sites per sequence.
  $N_{rem} \leftarrow 1 \times p_{N_m}$ vector filled with N_s ▷ Initialize the N_{rem} vector.
  $C_{part} \leftarrow 1 \times p_{N_m}$ vector filled with 1 ▷ Initialize the combinations vector.
  for $n_j = 1, 2, ..., n_{max}$ **do**
    $C_{tmp} \leftarrow 1 \times \max \boldsymbol{M_{part}}$ vector filled with 1 ▷ A maximum of $\max \boldsymbol{M_{part}}$ different a_j's.
    $C_j \leftarrow \binom{L_s - n_j L_m + n_j}{n_j}$ ▷ Compute all ways n_j sites can be aligned in a sequence.
    for $l = 1, 2, ..., p_{N_m}$ **do**
      $a_j \leftarrow \boldsymbol{M_{part}}(l, n_j)$ ▷ Read an element from $\boldsymbol{M_{part}}$.
      if $a_j > 0$ **then** ▷ Ignore elements that do not contribute to C_p.
        if $C_{tmp}(a_j) = 1$ **then** ▷ Only compute combinations if necessary.
          $C_{tmp}(a_j) \leftarrow C_j^{a_j}$
        end if
        $C_{part}(l) \leftarrow C_{part}(l) \cdot \binom{N_{rem}}{a_j} \cdot C_{tmp}(a_j)$ ▷ Multiply in the contribution from the current a_j.
        $N_{rem}(l) \leftarrow N_{rem}(l) - a_j$ ▷ Remove current a_j in preparation for the next a_j.
      end if
    end for
  end for
  for $l = 1, 2, ..., p_{N_m}$ **do** ▷ Sum up the contributions from all partition sets.
    $C_{p,ANR} \leftarrow C_{p,ANR} + C_{part}(l)$
  end for
  return $C_{p,ANR}$
end procedure

TABLE IV
EXAMPLE PARTITIONING MATRIX $\boldsymbol{M_{part}}$ FOR $N_m = n_{max} = 5$.

Partition	Sites per sequence, n_j				
number, l	1	2	3	4	5
1	0	0	0	0	1
2	1	0	0	1	0
3	0	1	1	0	0
4	2	0	1	0	0
5	1	2	0	0	0
6	3	1	0	0	0
7	5	0	0	0	0

number. There are many redundant numbers in a partition set, as can be seen in our example in Table IV: The binomial coefficients and $C_{p,many}$ components are computed multiple times for $a_j = 1$ and $n_j = 1$, $n_j = 2$, and $n_j = 1$. In this small example, these binomials and $C_{p,many}$ components are computed 12 times each, versus the actual 10 unique binomial coefficients and 9 $C_{p,part}$'s. When N_m increases, this difference becomes larger. For instance, 97 $C_{p,many}$ components must be computed when using $N_m = 10$, while only 26 are strictly required.

We propose an improved algorithm to compute $C_{p,ANR}$, shown in Algorithm 1. The algorithm basically computes the $C_{p,part}$ components generically for each unique n_j, and then raised to the appropriate power for each unique $\{a_j, n_j\}$ pair. The $\binom{N_{rem}}{a_j}$ components are computed for each $\{N_{rem}, a_j\}$ pair, and N_{rem} is adjusted for each non-zero element in each partition l.

III. EVALUATION

The number of combinations possible with an ANR alignment, $C_{p,ANR}$, is always at least as large as the number of possible combinations when using ZOOPS. The lowest number of combinations using ANR is when there is room for at most one site per sequence, i.e. when

$$N_{max} = \left\lfloor \frac{L_s}{L_m} \right\rfloor = 1. \quad (16)$$

When (16) is satisfied, the alignment becomes a ZOOPS alignment.

Assuming that $N_{max} \leq 2$, the ratio of combinations $C_{p,ANR}/C_{p,ZOOPS}$ will depend on L_s, N_s, and n_{max}. In order to illustrate how the ratio of $C_{p,ANR}/C_{p,ZOOPS}$ changes, consider the following four examples. Assume a motif of length $L_m = 40$ in all four examples.

When computing p-values for an alignment, it can be useful to adjust the number of sites, N_m, and select the alignment which meets some predetermined E-value threshold. In the first two examples, the data set consists of a fairly small number of gene sequences with $N_s = 10$. Since ZOOPS is only valid for $2 \leq N_m \leq N_s$, no N_m larger than 10 is considered. Adjusting the length of the gene sequences, L_s, from 100 to 300 the ratios for each L_s are shown in Fig. 10.

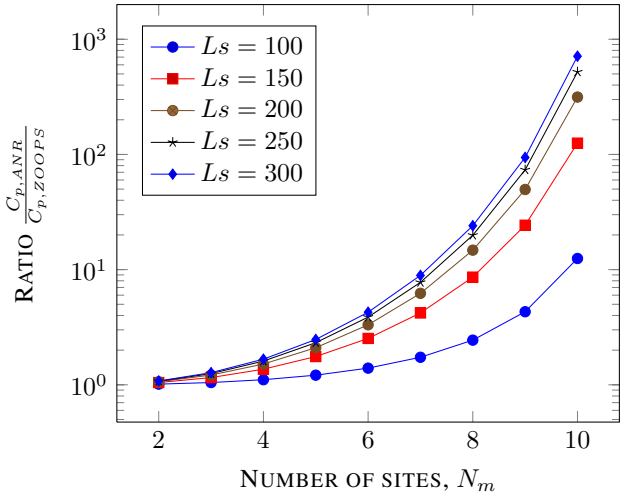

Fig. 10. Ratio of $C_{p,ANR}$ vs. $C_{p,ZOOPS}$ for $N_s = 10$ and $L_m = 40$.

It is clear from Fig. 10 that increasing the number of sites to align leads to a significantly larger number of possible combinations for ANR than for ZOOPS; up to several orders of magnitude. The ratio also increases with larger N_s, but the increase will taper off when the length of the sequences are long enough to fit all N_m sites in each, i.e. when $n_{max} = N_m$.

In the next example, the length of the sequences is set to be $L_s = 100$, while the length of the motif is changed from 10 to 50 nucleotides. The results can be seen in Fig. 11. Here, $L_m = 10$ represents the upper bound, since all 10 of the sites may fit inside a single sequence, while $L_m = 50$ is close to the lower bound since only two sites will fit in a sequence. Again, the trend of increasing ratio with increasing N_m is clear, and more pronounced the larger N_{max} becomes.

In the next two examples, the number of sequences in the gene set has been increased to $N_s = 100$. The third example uses the same L_s values as the first example, but since there are more sequences to align in, the number of motifs has been increased to span from $N_m = 2$ up to 20. The results are presented in Fig. 12. The trend of the ratio between ANR and ZOOPS is much more subdued here than in the first two examples. The reason for this is due to the increasing influence of the $\binom{L_s - n_j L_m + n_j}{n_j}^{a_j}$ coefficients versus the effect of the integer partitioning contributions.

In the last example, the length of the sequence has again

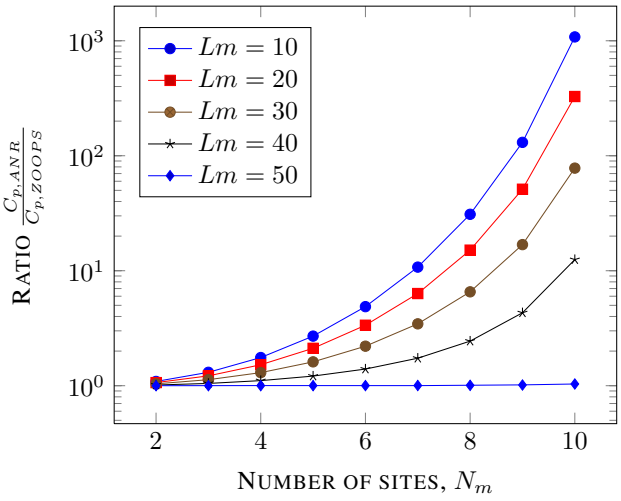

Fig. 11. Ratio of $C_{p,ANR}$ vs. $C_{p,ZOOPS}$ for $N_s = 10$ and $L_s = 100$.

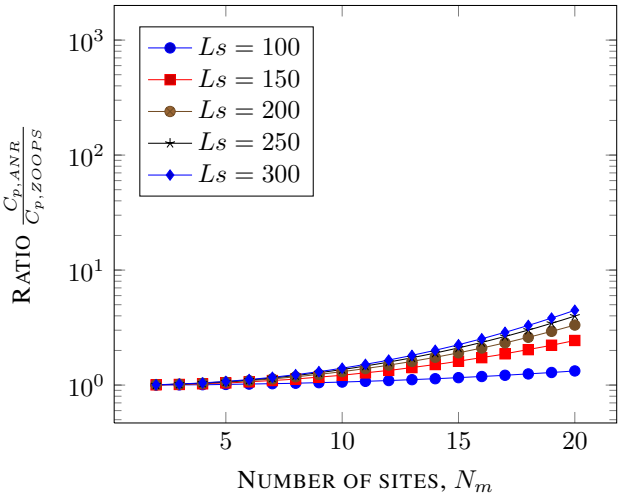

Fig. 12. Ratio of $C_{p,ANR}$ vs. $C_{p,ZOOPS}$ for $N_s = 100$ and $L_m = 40$.

been fixed to $L_s = 100$, while the length of the motif is adjusted. The results are shown in Fig. 13. As with the previous example, the increase in the $C_{p,ANR}/C_{p,ZOOPS}$ ratio is subdued. Again, the influence of the integer partitioning has a minor role compared to the increase in the number of sequences in the data set.

IV. Conclusion and Discussion

In this paper we have explored the combinatorial aspects of genetic motif alignments. We have shown that using a p-value to compute an E-value for a motif alignment can lead to an E-value that diverges significantly from the correct value if the alignment model used in the calculations does not correspond with the actual motif alignments. E.g., using a zero or one occurrence per sequence (ZOOPS) alignment model for an alignment which is actually an any number of repetition (ANR) model.

We have shown the mathematical theory behind the computation of the accurate number of alignment possibilities

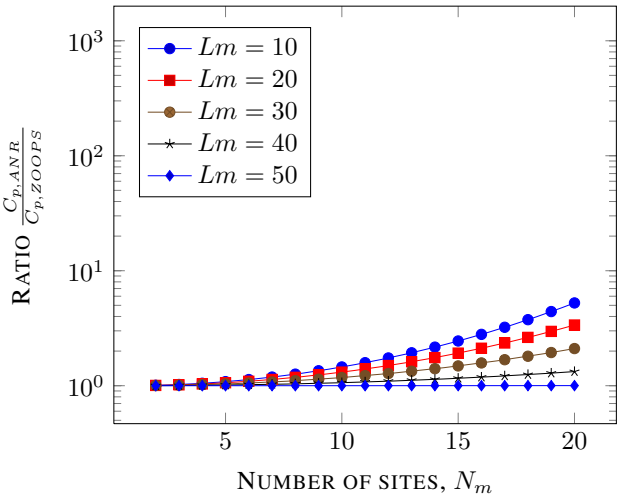

Fig. 13. Ratio of $C_{p,ANR}$ vs. $C_{p,ZOOPS}$ for $N_s = 100$ and $L_s = 100$.

for each of the different alignment models, such as one occurrence per sequence (OOPS), ZOOPS, and ANR, and we have also suggested an algorithm for computing the number of combinations for the ANR model. We have also compared the total number of combinations for the cases of ZOOPS and ANR for various example scenarios, and found that when the number of motif match sites is within the same range as the number of sequences there is a large difference between the numbers produced for the ZOOPS and ANR models. In those cases where the data sets has a much larger number of sequences than match sites in the motif alignment, the difference between ZOOPS and ANR is sufficiently low that using the simpler ZOOPS method for computing the number of combinations will not lead to a significant difference from the true number of combinations for ANR.

REFERENCES

[1] M. Tompa, N. Li, T. L. Bailey, G. M. Church, B. De Moor, E. Eskin, A. V. Favorov, M. C. Frith, Y. Fu, W. J. Kent, V. J. Makeev, A. A. Mironov, W. S. Noble, G. Pavesi, G. Pesole, M. Regnier, N. Simonis, S. Sinha, G. Thijs, J. van Helden, M. Vandenbogaert, Z. Weng, C. Workman, C. Ye, and Z. Zhu, "Assessing computational tools for the discovery of transcription factor binding sites." *Nat Biotechnol*, vol. 23, no. 1, pp. 137–144, Jan 2005.

[2] T. A. Down and T. J. P. Hubbard, "Nestedmica: sensitive inference of over-represented motifs in nucleic acid sequence." *Nucleic Acids Res*, vol. 33, no. 5, pp. 1445–1453, 2005.

[3] T. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learning*, vol. 21, no. 1-2, pp. 51–80, Oct-Nov 1995.

[4] G. Pavesi, G. Mauri, and G. Pesole, "An algorithm for finding signals of unknown length in dna sequences." *Bioinformatics*, vol. 17 Suppl 1, pp. S207–14, 2001.

[5] G. Z. Hertz and G. D. Stormo, "Identifying dna and protein patterns with statistically significant alignments of multiple sequences." *Bioinformatics*, vol. 15, no. 7-8, pp. 563–577, Jul-Aug 1999.

[6] C. Pizzia and E. Ukkonen, "Fast profile matching algorithms - a survey," *Theoretical Computer Science*, vol. 395, no. 2-3, pp. 137–157, MAY 1 2008.

[7] N. Nagarajan, P. Ng, and U. Keich, "Refining motif finders with e-value calculations," *Regulatory Genomics*, p. 73, 2006.

[8] G. Andrews, *The theory of partitions*. Cambridge University Press, 1998.

[9] G. Andrews and K. Eriksson, *Integer partitions*. Cambridge University Press, 2004.

[10] A. Zoghbi and I. Stojmenovic, "Fast algorithms for generating integer partitions," *International Journal of Computer Mathematics*, vol. 70, no. 2, pp. 319–332, 1998.