# Sharper Upper and Lower Bounds for an Approximation Scheme for Consensus-Pattern

Broňa Brejová Daniel G. Brown Ian M. Harrower Alejandro López-Ortiz
Tomáš Vinař

School of Computer Science
University of Waterloo
{bbrejova,browndg,imharrow,alopez-o,tvinar}@cs.uwaterloo.ca

**Abstract**

We present sharper upper and lower bounds for a known polynomial-time approximation scheme due to Li, Ma and Wang [7] for the Consensus-Pattern problem. This NP-hard problem is an abstraction of motif finding, a common bioinformatics discovery task. The PTAS due to Li *et al.* is simple, and a preliminary implementation [8] gave reasonable results in practice. However, the previously known bounds on its performance are useless when runtimes are actually manageable. Here, we present much sharper lower and upper bounds on the performance of this algorithm that partially explain why its behavior is so much better in practice than what was previously predicted in theory. We also give specific examples of instances of the problem for which the PTAS performs poorly in practice, and show that the asymptotic performance bound given in the original proof matches the behaviour of a simple variant of the algorithm on a particularly bad instance of the problem.

## 1 Introduction

Bioinformaticists often find themselves with several different DNA or protein sequences that are known to share a particular function, but where the origin of the function in the sequence is unknown. For example, suppose one has the DNA sequence of the region surrounding several genes, known to be regulated by a particular transcription factor. Here, the shared regulatory behavior may be caused by a sequence element common to all, to which the transcription factor binds. Discovering this experimentally is very expensive, so computational approaches can be helpful to limit searches.

The motif discovery problem is an abstraction of this problem. In it, we are given $n$ sequences, all of length $m$, over an alphabet $\Sigma$. We seek a single motif, of length $L$ that is found approximately as a substring of all sequences. Several variants of this problem exist. One can seek to minimize the maximum Hamming distance between the motif and its instances in all strings (*e.g.* [10, 2]), maximize the information content (minimize the entropy) of the chosen motif instances (*e.g.* [3, 1, 6]), or minimize the total of the Hamming distances between the motif and its instances [7]. This latter problem can be formally defined as follows:

**Definition 1 (Consensus-Pattern).** *Given: $n$ sequences $s_1, \ldots, s_n$, each of length $m$ and over an alphabet of size $A$. Find a substring $t_i$ of given length $L$ in each of the sequences and a median string $s$ of length $L$ so that the total Hamming distance $\sum_i d_H(s, t_i)$ is minimized.*

| Condition | New results | | Previous upper bound |
|---|---|---|---|
| | Lower bound | Upper bound | |
| $r = 1$ | 2 | 2 | N/A |
| $r = 3$ | 1.5 | $\approx 1.528$ | $\approx 1 + 4.006 \cdot (A - 1)$ |
| general $r$<br>binary alphabet | $1 + \Theta(1/r^2)$<br>conjecture: $1 + \Theta(1/\sqrt{r})$<br>(proved for sampling without replacement) | | $1 + \Theta(1/\sqrt{r})$ |
| general $r$<br>general alphabet | $1 + \Theta(1/r^2)$<br>conjecture: $1 + \Theta(1/\sqrt{r})$<br>(proved for sampling without replacement) | | $1 + \Theta(A/\sqrt{r})$ |

Table 1: **Overview of the results.**

Li, Ma and Wang [7] give a very simple polynomial-time approximation scheme (PTAS) for this combinatorial motif problem. For a given value of $r$, consider all choices of $r$ substrings of length $L$ from the $n$ sequences. We note explicitly here that the sampling is made with replacement, so that the same substring may occur multiple times. For each such collection $\mathcal{C}$ of substrings, we compute its consensus by identifying the most common letter in the first position of each chosen substring, the second position, and so on, producing a motif $M_{\mathcal{C}}$. It is easy to identify for a given motif $M_{\mathcal{C}}$ its closest match in each of the $n$ sequences, and thus its score. We do this for all $n^r(m - L + 1)^r$ possible collections of $r$ substrings, and pick the collection with the best score. The algorithm has $O(L(nm)^{r+1})$ run time, and is thus polynomial-time for any particular value of $r$. Li *et al.* also give an upper bound on the worst-case approximation ratio of this algorithm for $r \geq 3$:

$$1 + \frac{4A - 4}{\sqrt{e}\left(\sqrt{4r + 1} - 3\right)}, \tag{1}$$

where $A$ is the size of the alphabet $\Sigma$. For example, if $r = 3$, this approach gives an algorithm that runs in $O(L(nm)^4)$ runtime, but whose approximation guarantee for DNA sequences (where $A = 4$) is approximately 13. To achieve a reasonable approximation ratio, 2, we would have to use $r \geq 8$ for DNA sequences, or $r \geq 27$ for protein sequences ($A = 20$), giving hopelessly large running times. The high value of the proven bound would seem to suggest that the algorithm will be useless in practice.

However, many successful combinatorial motif finders do work by generalizing from small samples in this way, such as SP-STAR [10] and CONSENSUS (samples of 1) [3], COMBINE (samples of 2 to 3) [9], COPIA (samples of arbitrary size) [8]. Here, focusing on Li *et al.*'s PTAS, we show tighter bounds on its performance much closer to reasonable numbers for practical values of $r$. We also provide the first substantial lower bounds on the PTAS's performance, by identifying specific examples of the problem for which the algorithm performs poorly. In the general case, for a binary alphabet, we find that the variant of the algorithm that works by sampling *without* replacement performs poorly on a particular bad example, and we conjecture that our example will also be difficult for the original Li *et al.* algorithm that samples *with* replacement.

Our results are summarized in Table 1.

## 2 Basic Observations

We begin our discussion of the algorithm by noting that it is sufficient to look at the performance of the PTAS when run on the actual instances of the motif (which are sequences of length $L$), rather than on the $m$-letter input strings.

**Lemma 1.** *Suppose that the PTAS of Li* et al. *achieves approximation ratio $\alpha$ for a given set $s_1 \ldots, s_n$ of input sequences, motif length L and sample motif size $r$. Suppose also that the instance of the optimal motif in sequence $s_i$ is $t_i$. Then the PTAS, if run only on the sequences $t_1, \ldots, t_n$, would achieve approximation ratio at least $\alpha$.*

*Proof.* We begin by noting that if $m = L$, the actual problem is trivial: the optimal motif $s^*$ is the consensus of all of the input strings.

However, the PTAS still is well defined in this case, even though the actual optimization problem is trivial. It examines all sets $\mathcal{C}$ of $r$ strings, including ones where the same string is chosen multiple times, and for each of them, computes its consensus $M_{\mathcal{C}}$. Then, the central motif $M_{\mathcal{C}^*}$ with smallest total Hamming distance to all $s_i$ is chosen as the motif center.

This motif center can be no better than the one found by the PTAS when run on the entire $m$-letter strings, because the set of substrings we have considered in the truncated problem is a subset of the set of substrings we would have examined in the full problem. As such, if the original algorithm would have found a solution whose approximation ratio is $\alpha$, we can only have done as well or worse in the truncated problem. $\square$

This lemma is useful because if we can show that, for given values of $L$, $n$ and $r$, and when run only on the optimal motif instances, the PTAS has approximation ratio at most $\beta$, then its approximation ratio on longer strings can still be no worse than $\beta$.

To simplify notation, we assume that the alphabet is $\{0, 1, \ldots, A-1\}$. In the special case we focus on, where $m = L$, we also always renumber the characters in each column, so the consensus for that column is 0. This causes the overall optimal motif to be $s^* = 0^L$. This transformation only works when $m = L$; it does not work when $m > L$.

Finally, we can encounter the problem of ties, that is, a situation when the consensus string $u$ of some collection $\mathcal{C}$ is not unique. Consider for example $r = 3$ and input strings `01`, `02`, `10`, and `20`. The optimal motif is `00`, with cost 4. If $\mathcal{C}$ contains the first three strings, the consensus $M_{\mathcal{C}}$ can be any of the strings `00`, `01`, and `02`. The first of them is optimal, but the latter two have cost 5.

It is not realistic to assume that the PTAS will always guess the best of all possible consensus strings; their number can be exponential in $L$. For simplicity, we assume that the PTAS will choose the worst consensus $u$ out of all possibilities, and study the performance of this "unlucky" motif finding algorithm, which in our example would choose either `01` or `02`.

## 3 Upper Bounds

In this section, we give better worst-case bounds on the approximation guarantee of the algorithm in the cases where $r = 1$ or $r = 3$, corresponding to algorithms with quadratic or quartic bounds on their runtime.

**Theorem 1.** *The approximation ratio of the PTAS is at most 2 for all values of $r$, including $r = 1$, and for any alphabet size $A$.*

*Proof.* Let $c$ be the cost of the optimal motif $0^L$, that is, the total number of non-zero elements in all sequences. Let $a_i$ be the number of non-zero elements in sequence $s_i$. If the PTAS chooses sequence $s_i$ as the motif (which will happen when the $r$ samples from the $n$ sequences are all of $s_i$), the cost will increase by at most $n$ for every column where $s_i$ has non-zero element. Therefore the cost will be at most $c + na_i$. When we sum this quantity over all sequences $s_i$, we get $nc + n\sum_{i=1}^{n} a_i = 2nc$. Since the sum of costs for $n$ different potential motifs $s_i$ is at most $2nc$, at least one of these has cost at most $2c$, which means the approximation ratio is at most 2. □

**Theorem 2.** *The approximation ratio of the PTAS for $r = 3$ is at most $(64 + 7\sqrt{7})/54 \approx 1.528$ regardless of the size of the alphabet.*

*Proof.* Let $p$ be the proportion of zeroes and $q = (1-p)$ be the proportion of non-zeroes in the input sequences. The optimal cost is therefore $qnL$. Let $b_j$ be the number of non-zeroes in column $j$.

The algorithm will examine all possible samples consisting of 3 rows, choosing the one with the best consensus string. To get an upper bound, we will consider an expected cost of consensus string obtained by sampling 3 rows uniformly at random.

For each column, we can estimate the expected cost of the column. The non-zero consensus can be achieved in a particular column only if two or three of the rows containing non-zero entries are chosen. If the column contains $b$ non-zero entries, there are $b^3 + 3b^2(n-b)$ such samples. Each of these samples will incur cost of at most $n$ in this column. The consensus will be zero for samples with two or three zeroes (their number is $(n-b)^3 + 3(n-b)^2b$). Each of these samples will incur cost $b$ in this column.

Thus the expected cost $E(b)$ for a column with $b$ non-zeroes is at most $C(b)/n^3$, where $C(b)$ is the sum of costs over all triples of rows:

$$C(b) = [b^3 + 3b^2(n-b)]n + [(n-b)^3 + 3(n-b)^2b]b = 2b^4 - 5b^3n + 3b^2n^2 + bn^3. \tag{2}$$

From linearity of expectation, the expected cost over all columns is

$$E(b_1, \ldots, b_L) = \sum_{j=1}^{L} E(b_j) = \frac{1}{n^3} \cdot \sum_{j=1}^{L} C(b_j). \tag{3}$$

There must exist a sample with cost at most $E(b_1, \ldots, b_L)$. Such a sample achieves approximation ratio $E(b_1, \ldots, b_L)/qnL$.

We will prove by induction on $L$ that $E(b_1, \ldots, b_L) \leq HqnL$, where $H = (64 + 7\sqrt{7})/54$. This implies that $H \approx 1.528$ is an upper bound on the approximation ratio for $r = 3$.

For $L = 1$, the approximation ratio is

$$E(qn)/qnL = 2q^3 - 5q^2 + 3q + 1. \tag{4}$$

The maximum of this ratio, which is equal to $H$, is reached when $q = \frac{5-\sqrt{7}}{6}$.

Now, assume that the induction hypothesis is true for $L-1$. We will prove that it is also true for $L$. The expected cost of the first column is $E(b_1)$, which can be computed with Equation 2 above. By our induction hypothesis, the expected cost of the remaining $L-1$ columns is at most $(qnL - b_1) \cdot H$. Note that $qnL - b_1$ is the optimal cost for the remaining $L-1$ columns. Therefore:

$$E(b_1, \ldots, b_L) \leq E(b_1) + (qnL - b_1) \cdot H = \underbrace{\frac{2b^4 - 5b^3n + 3b^2n^2 + (1-H)bn^3}{n^3}}_{(*)} + HqnL \tag{5}$$

We want to prove, that (*) is never positive for $b$ in the range $0 \le b \le n$. Indeed, (*) can be simplified as $(b/(108n^3)) \cdot (6b - (5 + 2\sqrt{7})n) \cdot (6b - (5 - \sqrt{7})n)^2$. The first and third factors are always non-negative, and the second factor non-positive for all $b < n$. Therefore the whole term (*) is never positive on the interval. □

It is, in fact, possible to easily characterize the "worst-case" scenario that maximizes $E(b_1, \dots, b_L)$: this is achieved when the non-zero elements are distributed equally among a subset of the columns as follows.

**Lemma 2.** *For a given $q$, $n$, and $L$, $E(b_1, \dots, b_L)$ is maximized, when for some $k \le L$, $b_1, \dots, b_k = 0$, and $b_{k+1} = b_{k+2} = \dots = b_L \le n$ (if we allow $b_1, \dots, b_L$ to be non-integral).*

*Proof.* (by induction on $L$). For $L = 1$, the hypothesis holds trivially.

Let us assume that the hypothesis holds for all $L' < L$. Without loss of generality, we assume that the columns are sorted by $b_j$. If $b_1 = 0$, the hypothesis holds trivially from induction hypothesis. Let $b_1 > 0$. Then, by induction hypothesis, all the rest of the columns must by distributed equally (there are no columns with $b_i = 0$, since $b_1$ is the smallest). The cost will be therefore:

$$C(b_1) + (L-1) \cdot C\left(\frac{qnL - b_1}{L-1}\right), \tag{6}$$

where $nL(q-1) + n \le b_1 \le qn$, and $b_1 > 0$. This is indeed maximized for $b_1 = qn$, as can be shown by straightforward algebraic manipulation. □

## 4 Lower Bounds

In this section, we present examples of inputs for which the Li *et al.* PTAS performs poorly. These examples give lower bounds on the approximation guarantee. For small values of $r$, we are able to give lower bounds which almost match our upper bounds the from previous section. For general values of $r$, we show an example where the PTAS has approximation ratio $1 + \Theta(1/r^2)$. Finally, we conjecture that lower bound on approximation ratio matches asymptotically the upper bound $1 + \Theta(1/\sqrt{r})$; to support this claim, we present an example for which a slightly modified algorithm has approximation ratio at least $1 + \Theta(1/\sqrt{r})$.

**Theorem 3.** *For $r = 1$, the approximation ratio is at least 2, even for binary inputs.*

*Proof.* We set $L = n$. The input will be the $n \times n$ identity matrix $I_n$, with ones on the diagonal and zeroes everywhere else. The cost of the optimal solution is $n$. The result of the PTAS for $r = 1$ will be one of the matrix rows, with cost $2n - 2$. The approximation ratio is therefore $2 - 2/n$, which converges to 2 as $n$ grows without bound. This shows that the upper bound 2 is tight for $r = 1$. □

**Theorem 4.** *For $r = 3$, the approximation ratio is at least $3/2$.*

*Proof.* For given $k$, consider the input with $n = 2k$, $L = 2$ containing for every $i = 1, 2, \dots, k$ strings $0i$ and $i0$. For example for $k = 2$ the input will be the following:

$$\begin{array}{cc} 0 & 1 \\ 0 & 2 \\ 1 & 0 \\ 2 & 0 \end{array}$$

The optimal solution is 00, with cost $2k$. However, assuming that the PTAS breaks ties in the worst possible way, it will find motif $0x$ or $x0$, with cost $3k-1$. $\square$

**Theorem 5.** *The approximation ratio of the PTAS is at least* $1+\Theta(1/r^2)$.

*Proof.* For any odd $r$, we create $n = r + 2$ sequences, each of length $L = (r + 5)/2$. The first $L-1$ columns of the first $L-1$ sequences will be an inverted identity matrix, with zeroes on the diagonal and ones everywhere else. The last column of these sequences contains zeroes. The remaining $n - L + 1$ sequences have zeroes in the first $L-1$ columns and one in the last column. For example for $r = 5$ we get the following input:

$$\begin{matrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{matrix}$$

The optimal solution $0^L$ has cost $c = (r + 1)(r + 5)/4$. Any solution that has a single one in it will have cost $c+1$, and it is clear that by choosing the last row $r$ times, the algorithm will find a motif at least as good as $0^{L-1}1$. We show that the PTAS cannot find the optimal solution.

Assume that the PTAS can obtain the optimal solution $0^L$. Then there must be some collection $\mathcal{C}$ of strings such that each column has more than $r/2$ zeroes. In particular, for the last column, more than half of these strings are chosen from the first $L-1$ sequences of the input. Thus, to achieve more than $r/2$ zeroes in any other column $i < L$, we have to include at least one copy of sequence $i$ (less than $r/2$ copies of the last $n - L + 1$ sequences are included). That means we need to include each of the first $L-1$ sequences. But then all of the first $L-1$ columns contain at least $L-2 = (r+1)/2$ ones, and we will not find the correct motif. This is a contradiction. Therefore the PTAS cannot achieve the optimal solution.

Therefore the approximation ratio is $(c+1)/c = 1 + 4/[(r+1)(r+5)] = 1 + \Theta(1/r^2)$. $\square$

We were unable to obtain a lower bound matching the original upper bound of $1 + \Theta(A/\sqrt{r})$ of Li *et al.* [7]. However, we conjecture this upper bound is tight; to support this conjecture, we offer a lower bound of $1 + \Theta(1/\sqrt{r})$ for a modified algorithm.

In the original PTAS, the samples from which the possible motifs are generated are performed *with* replacement, so that a given substring can be chosen multiple times. Here, we will consider a variation of this, which requires that the same substring be chosen only once. In our context, where $L = m$, this means we will choose all subsets of size $r$ of the $n$ input strings, compute their consensus sequences, and take the best of these possible choices. Note that this modified algorithm will always give the same or worse results as the original PTAS.

We conjecture that this algorithm also forms a PTAS with similar bound to the original, and we conjecture that bad examples of the problem for this algorithm will also be bad examples of the problem for the Li *et al.* PTAS, with similar lower bounds.

**Theorem 6.** *Consider a modification of the PTAS, where we allow only a single sample from each input sequence. Such modified algorithm has approximation ratio at least* $1 + \Theta(1/\sqrt{r})$*, even for binary alphabet.*

*Proof.* We will give instances that give this bound in the limit as $r$ goes to infinity. Let $r$ be of the form $(2k+1)^2$, and let $n = 2r$. Our problem instance will have $L = \binom{2r}{r+\sqrt{r}}$ columns — all possible columns with $r - \sqrt{r}$ ones and $r + \sqrt{r}$ zeros. The optimal solution $0^L$ will have score $L \cdot (r - \sqrt{r})$.

The modified PTAS will examine all possible subsets of $r$ sequences (sampling without replacement). Note that in this particular example, any combination of $r$ rows will give rise to a consensus string with the same cost. This is because any combination of $r$ rows can be transformed to any other such combination by rearranging columns. Therefore every combination gives a consensus string with the same number of ones.

Since all samples are equivalent, we could as easily study a random sample of size $r$ chosen from the $2r$ sequences, and identify the expected number of ones in the consensus string. Considering a single column, let $p_r$ be the probability that the consensus for this column will be one, i.e. the random sample without replacement of size $r$ from the population of $r - \sqrt{r}$ ones and $r + \sqrt{r}$ zeros will contain more than $r/2$ ones. By linearity of expectation, the expected number of ones in the consensus string will then be $L \cdot p_r$.

Since the symmetry argument shows that all solutions have the same value, all samples will identify a consensus string with $L \cdot p_r$ ones, and so will the algorithm. Thus, the modified PTAS will give a solution with value $L \cdot p_r \cdot (r + \sqrt{r}) + L \cdot (1 - p_r)(r - \sqrt{r})$. Since the optimum has value $L \cdot (r - \sqrt{r})$, the approximation ratio is $1 + \frac{2p_r\sqrt{r}}{r-\sqrt{r}} > 1 + 2p_r/\sqrt{r}$. Thus, if we can show that, as $r \to \infty$, $p_r$ is greater than some constant $\varepsilon$, this will suffice to prove that the algorithm has approximation ratio at least $1 + \Theta(1/\sqrt{r})$.

We prove this by using the Central Limit Theorem for Finite Populations (*e.g.* [11, Section 3.4]). This is the variation on the Central Limit Theorem for sampling from finite populations without replacement. Specifically, it implies that if we sample $r$ times from a pool of $r + \sqrt{r}$ zeros and $r - \sqrt{r}$ ones, then as $r \to \infty$, the number of ones picked converges to a normal distribution, with mean $\mu = 1/2(r - \sqrt{r})$ and variance $\sigma^2 = r/8 - 1/8 \geq r/16$.[1]

We are interested in probability $p_r$ that the number of ones in the sample is at least $r/2$. Note, that in the normal distribution $N(\mu, \sigma)$, $r/2 \leq \mu + 2\sigma$, and therefore

$$p_r \geq \Pr(N(\mu, \sigma) \geq \mu + 2\sigma) = \Pr(N(0,1) \geq 2) \approx 0.023,$$

as $r \to \infty$. Therefore $p_r$ has a constant lower bound, which is what we wanted to show. □

The expected cost of the consensus can be computed using similar method for the original PTAS as well. However, the symmetry argument does not hold any more, and therefore there can be samples with cost lower than expected cost. Thus the proof presented above cannot be directly extended to the original PTAS.

## 5 Conclusion and open problems

We have given lower and upper bounds for the performance of an extremely simple polynomial-time approximation scheme due to Li *et al.* [7] for the CONSENSUS-PATTERN problem, which is an abstraction of a common biological sequence motif detection problem. The PTAS examines all

[1] The technical condition for the theorem to hold requires

$$\lim_{N \to \infty} \frac{M(N-M)S(N-S)}{N^3} = \infty,$$

where $N$ is the size of the population, $M$ is the number of ones in the population, and $S$ is the size of the sample. This condition holds trivially in our case.

choices of $r$ substrings of the input sequences, computes the consensus sequence of the substrings, and then finds the best matches to this consensus in all strings. After examining all possible choices, it chooses as the motif the consensus substring chosen with best overall performance.

Our bounds give a partial explanation for why algorithms based on sampling substrings of the input give good performance in practice. While they do not improve the upper bounds on the approximation ratio for large sample sizes, they do show that, for small sample sizes $r$, such as 3, the extremely simple PTAS can guarantee performance ratios of at most 1.528, as compared with bounds much larger with the original Li *et al.* proof.

We have also given new lower bounds on the best possible approximation ratio of the PTAS, by showing examples for which the PTAS has poor performance. In the case of 1-substring samples, our bad example gives approximation ratio converging to 2, which matches our upper bound. In the case of 3-substring samples, we show that the best approximation ratio is at least 1.5, which is very close to our upper bound of 1.528. In the more general case of a binary input alphabet and arbitrary sample size $r$, we show an instance with lower bound $1 + \Theta(1/r^2)$. We also show that the slight variation on the PTAS that does not allow sampling with replacement, but only *without* replacement can only achieve ratios of at least $1 + \Theta(1/\sqrt{r})$, by applying limit theorems for samples of finite populations. We conjecture that this bound, which asymptotically matches the proven upper bound due to Li *et al.* for the original PTAS, also applies when sampling is allowed with replacement.

We should note that our worst-case bounds may have little applicability to real instances of motif-finding problems in practice. Indeed, in a quite different direction than we have gone in this work, many authors (*e.g.* [4, 5]) have focused on probabilistic models of sequences, and on the information content of subtle motifs, to identify the probability that a particular algorithm will correctly identify a motif implanted in them. In particular, these authors have focused on the probability of identifying weak motifs, whose score is not much higher than "decoys" in the sequence. Our results, which show that decoys are certain to be found by the PTAS, are for similarly weak motifs, but with no probabilistic basis.

**Open problems** Numerous open problems still remain in this area. We would be interested in developing sharper bounds for the case of larger input alphabets. While our bounds on binary alphabets naturally carry to the case of larger alphabets, the upper bound grows with the size of the alphabet. It is possible that the true upper bound does not depend on the alphabet size, rather than that the current lower bound is too small. This is also supported by our upper bound for $r = 3$ of 1.528 regardless of the size of the alphabet (in fact, this upper bound also holds for all values of $r$ greater than 4).

The other open problem we would suggest is to determine whether the algorithm based on sampling without replacement needed for the proof of Theorem 6 can be proven to be a PTAS with the same guarantee, or whether our bad example or one like it can be used to prove an analogous lower bound for the original PTAS.

**Acknowledgements** All authors are supported by the National Science and Engineering Research Council of Canada. Additionally, the second author is supported by the Human Frontier Science Program. We would like to thank Christopher Small and Mary Thompson for pointing us to the presentation of the limit theorem used in the proof of Theorem 6 found in Dr. Thompson's book [11].

## References


[1] T.L. Bailey and C. Elkan. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. In *Proceedings of the 2nd International Conference on Intelligent Systems for Molecular Biology (ISMB 1994)*, pages 28–36. AAAI Press, 1994.

[2] J. Buhler and M. Tompa. Finding motifs using random projections. In *Proceedings of the 5th Annual International Conference on Computational Molecular Biology (RECOMB 2001)*, pages 69–76, 2001.

[3] G.Z. Hertz and G.D. Stormo. Identifying DNA and protein patterns with statistically significant alignments of multiple sequences. *Bioinformatics*, 15(7-8):563–577, 1999.

[4] U. Keich and P.A. Pevzner. Finding motifs in the twilight zone. *Bioinformatics*, 18:1374–1381, 2002.

[5] U. Keich and P.A. Pevzner. Subtle motifs: defining the limits of motif finding algorithms. *Bioinformatics*, 18:1382–1390, 2002.

[6] C.E. Lawrence, S.F. Altschul, M.S. Boguski, J.S. Liu, A.F. Neuwald, and J.C. Wootton. Detecting subtle sequence signals: a Gibb's sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.

[7] M. Li, B. Ma, and L. Wang. Finding similar regions in many strings. *Journal of Computer and System Sciences*, 65(1):73–96, 2002.

[8] C. Liang. COPIA: A New Software for Finding Consensus Patterns in Unaligned Protein Sequences. Master's thesis, University of Waterloo, October 2001.

[9] J. Liu. A Combinatorial Approach for Motif Discovery in Unaligned DNA Sequences. Master's thesis, University of Waterloo, March 2004.

[10] P.A. Pevzner and S. Sze. Combinatorial approaches to finding subtle signals in DNA sequences. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, pages 269–278, 2000.

[11] M.E. Thompson. *Theory of Sample Surveys*. Chapman and Hall, 1997.