Gapped consensus motif discovery: evaluation of a new algorithm based on local multiple alignments and a sampling strategy

Christine Sinoquet

Gapped consensus motif discovery: evaluation of a new algorithm based on local multiple alignments and a sampling strategy

Christine Sinoquet

Laboratoire d'Informatique de Nantes-Atlantique (LINA)
2, rue de la Houssinière - BP 92208, 44322 Nantes Cedex 3 - France

— Combinatoire et Bio-informatique —

RESEARCH REPORT

N° 05.03

May 2005

Christine Sinoquet

Gapped consensus motif discovery: evaluation of a new algorithm based on local multiple alignments and a sampling strategy

18 p.

Les rapports de recherche du Laboratoire d'Informatique de Nantes-Atlantique sont disponibles aux formats PostScript® et PDF® à l'URL :
http://www.sciences.univ-nantes.fr/lina/Vie/RR/rapports.html

Research reports from the Laboratoire d'Informatique de Nantes-Atlantique are available in PostScript® and PDF® formats at the URL:
http://www.sciences.univ-nantes.fr/lina/Vie/RR/rapports.html

Gapped consensus motif discovery: evaluation of a new algorithm based on local multiple alignments and a sampling strategy

Christine Sinoquet

sinoquet@lina.univ-nantes.fr

Abstract

We check the efficiency and faisability of a novel method designed for the discovery of *a priori* unknown motifs described as gaps alternating with specific regions. Such motifs are searched for as consensi of **non homologous** biological sequences. The only specifications required concern the maximal gap length, the minimal frequency for specific characters and the minimal percentage (quorum) of sequences sharing the motif. Our method is based on a cooperation between a multiple alignment method for a quick detection of local similarities and a sampling strategy running candidate position specific scoring matrices to convergence. This rather original way implemented for converging to the solution proves efficient both on simulated data, gapped instances of the so-called challenge problem, promoter sites in Dicot plants and transcription factor binding sites in *E.Coli*. Our algorithm compares favorably with the MEME and STARS approaches in terms of accuracy.

1 Introduction

A central task in post-genomics is automatic retrieval of a consensus motif in a set of biological sequences. We address a peculiar instance of the pattern discovery problem in a set S of n nucleotide sequences $\{s_1, s_2, \cdots, s_n\}$. In its general form, this problem may be stated as follows: identify a set of sub-word sets $\{SW_1, SW_2, \cdots, SW_n\}$ such that either $SW_i = \emptyset$ or $SW_i = \{o_{i1}, o_{i2}, \cdots, o_{ir_i}\}$ with any o_{ij} $(1 \leq j \leq ir_i)$ verifying a similarity constraint with any other o_{lp} $(1 \leq p \leq lr_l)$. In this paper we focus on the class of motifs allowing gaps, that is the class whose elements are of the type ACTGxxxxCTTxxGGxxxAAGA for example. The latter pattern contains three gaps with respective lengths 4,2 and 3. Motif characters differing from the wild-card character 'x' are the most frequent characters encountered over the set of occurrences of the consensus motif in the sequence set. They are called specific characters. We restrict to the case where each sequence contains at most one occurrence of the motif. An occurrence of the former pattern may be retrieved at position 4 in the following sequence: aataACGGgtggCGTaaGGtccAAGA. The number of mismatches *w.r.t.* the specific regions is 2. The notion of consensus motif is related to that of position specific scoring matrix (*pssm*), which represents a local alignment of sub-words of the same length. *pssm*$[c, j]$ yields the frequency of character c at position j over the considered sub-words.

Whatever the definitions of similarity, pattern type and response type (zero or one sub-word per sequence, exactly one sub-word, all possible sub-words), any instance of the *ungapped consensus motif retrieval* problem ($UCMR$) is NP-hard [1] and approximated algorithms based on scoring have to be designed for large datasets or long sequences. The *gapped consensus motif retrieval* problem ($GCMR$) is still more difficult, though of prime importance for users mining biological data. Here we study the faisability and the limits of a $GCMR$ algorithm whose principle is converging to the *pssm* solution through iterations confronting *pssm*$_s$ yielded by a $UCMR$ algorithm. First, we want to benefit from existing methods for local similarity search. *Ungapped multiple alignments* (UMA) are such algorithms. But experience on both simulated and biological data (**non homologous** sequences) teaches us that the less similar sequences, though known to share the motif, are uncorrectly aligned in an UMA performed on **the whole dataset**. Deriving the consensus motif from such a MA does not yield the optimal solution (see Figure 1 in Appendix for illustration). So we investigate a stochastic solution for the $GCMR$ problem, thus identifying and exploring a new direction of research. No *a priori* knowledge on the consensus motif is required except the maximal gap length for the consensus motif searched for. Neither do we require the user to specify a range for the number of gaps, nor do we even need information about the gap length range. Our proposition is half-way between pairwise sequence comparison and simultaneous comparison. The key idea rests on sampling the search space for motif candidates by confronting UMA_s built from small sequence datasets. To our knowledge, the idea of having a $UCMR$ method associated with a sampling strategy to perform $GCMR$ has not been explored yet. We first check the validity of the method for the highest frequency of specific characters, with datasets of size 40, sequences of length 50, motif length 15 containing either scattered short gaps or some rather long gaps. We also succeed in retrieving consensi for minimal frequency specifications under 100% in simulated datasets of size 100, sequence lengths of size 50 to 300, model lengths in the range $[14, 22]$ with different gap distributions. We retrieve binding sites in tandem with scattered gaps inserted in 50 artificial sequences of lengths 50 to 300. Finally we focus on 4 biological datasets (promoter sites in Dicot plants, transcription factor binding sites in $E.Coli$, sequence lengths ranging from 250 to 5800) and check that our method, compared with MEME and STARS, is as efficient in average.

The remainder of the paper is organized as follows. As preliminaries in section 1 we briefly categorize the main heuristics proposed in the literature for the $UCMR$ and $GCMR$ problems. We end this section introducing the terminology we use. In section 2, we outline our algorithm. Section 3 presents results obtained on simulated and biological data and discusses them. Further possible improvements are discussed in Conclusion.

2 Preliminaries

2.1 Related literature

Studying the $UCMR$ problem applied to realistic cases resorts to heuristics aiming at two goals: detection of characters which often occur simultaneously; optimization of an objective function modelling the divergence of

the occurrences "supporting" the consensus motif candidates. The commonly used objective functions are the χ^2 statistic and the log-likelihood ratio statistic [3]. A variant of the log-likelihood ratio is also known as the Kulback-Leibler divergence [8] or relative entropy. A recent detailed survey of the literature dedicated to $UCMR$ may be found in [12] or [6] for example. In the rest of this paragraph we cite a restricted number of approaches for the sake of succinctness.

Multiple alignment (MA) algorithms in the line of CLUSTALW[15] implement a bottom up pairwise sequence or alignment comparison. To identify characters frequently co-occurring at a constant distance, other methods consider different search spaces and exploring strategies. For the sake of simplicity in our explanations, we shall consider that the core algorithm of all methods described below searches for patterns of fixed length w. Except for CONSENSUS [7], all algorithms resort to sampling the search space. CONSENSUS systematically builds the *pssm*$_s$ of level l adding a sub-word from one of the sequences to each current multiple alignment of level $l-1$. The Expectation Maximization-based method MEME [4] estimates the probability that the shared motif starts in *any possible position* j for sequence i of the dataset, given the data and an initial guess at a description of the *pssm*. Then it reestimates the probability of nucleotides in the *pssm*. It runs such cycles until *pssm* convergence is reached. The Gibbs Sampler [9] drives the search relying on successive *guesses* at occurrence locations in the following way: given a set of current locations in all sequences but one, current *pssm* P and background frequencies B are calculated. Then every sub-word of length w in the excluded sequence is scored with a ratio of probabilities (generation with P/generation with B). Finally a location on the excluded sequence is chosen with a probability depending on the former ratio computed for the corresponding sub-word. Next cycle starts choosing at random the sequence to be excluded before performing one more step with the new location set. From the very start heuristic local alignment for pairwise comparison used seeds to detect potential similar regions (BLAST [2], FASTA [10], PatternHunter [11]). In the same line, the stochastic approach PROJECTIONS [5] is a heuristic designed for the $UCMR$ problem which iterates trials, each time sampling the search space with a different seed - a black mask of length w with only k windows (positions) to peer at the whole dataset-. All w-mers of the dataset are hashed to k-mers corresponding to their "k-projections". The key recording most entries is a hint to recover the potential consensus motif. In the MoDEL approach [6], a sub-optimal multiple alignment is obtained through exploring two neighborhoods of a current MA (a list of occurrence locations): the former being all MA_s shifted "a bit" from the current one; the latter consisting of all MA_s obtained with all possible location variations of a putative occurrence.

The still harder $UCMR$ topical subject is under work-in-progress in the STARS approach [12]. It identifies a potential common motif scanning the sequences one after another. It states as a first hypothesis that the first sequence is the common motif and performs motif splitting under a scoring scheme. It goes through several such cycles to escape the influence of the sequence scanning order.

2.2 Notations and definitions

The alphabet of the sequences is Σ.

Notation 1 $\forall x \in$ *alphabet* Σ, $1_{(x=y)} = 1$ *iff* $x = y$ *and* $1_{(x=y)} = 0$ *iff* $x \neq y$.

Before defining formally the consensus motif notion, let us first define the notion of *pssm* M relative to a set $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$ of n sequences.

Definition 2.1 *Given two integers* $minSeq$ *and* $minWidth$, *a* pssm M *associated with* $\mathcal{S}$ *is any matrix* $M \in \mathbb{R}^{+^{|\Sigma| \times l}}$ *with* $minWidth \leq l \leq \min_{s \in \mathcal{S}}\{length(s)\}$, *indexed with characters of* Σ, *which verifies:* $\exists\ su$ $(minSeq \leq su \leq n)$ *substrings* $o_i \in \Sigma^l$, *each included in one of the* n *sequences of* $\mathcal{S}$ *such that* $\forall c \in \Sigma, \forall\ j\ 1 \leq j \leq l$, $M[c,j] = \frac{1}{su} \sum_{i=1}^{su} 1_{(o_i[j]=c)}$.

In the following, we will say that $\mathcal{S}$ **supports** *the* pssm. *We will only consider* pssm$_s$ *supported by a minimum number of sequences:* $minSeq$ *at least.*

Notation 2 *We denote* M^+ *the vector* $\in \mathbb{R}^{+^l}$ *verifying:* $\forall\ j\ 1 \leq j \leq l,\ M^+[j] = max_{c \in \Sigma}\{M[c,j]\}$.

Notation 3 *The character in* Σ *corresponding to the greatest frequency at position* j *is denoted* $char(M^+[j])$.

Notation 4 *The notation $M^+{}_f[j]$ will bring conciseness to specify the following predicate: the value for the most frequent character at position j verifies $M^+[j] \geq f$. This character will be denoted $char(M^+{}_f[j])$.*

Notation 5 *Let $'x'$ ($\notin \Sigma$) be the wild-card character. We denote $mask(M)$ the string $\in \Sigma^l \cup \{'x'\}$ verifying:* $\forall\, j\ 1 \leq j \leq l,\ mask(M)[j] = \begin{cases} 'x' & if\ M^+[j] < f \\ char(M^+{}_f[j]) & otherwise. \end{cases}$

Definition 2.2 *Given $g, f \in \mathbb{N}$, we denote $maxGapLength_{g,f}(M)$ the following predicate: $max(j_2 - j_1 + 1 \in \mathbb{N} \mid \forall\, j\ 1 \leq j_1 \leq j \leq j_2 \leq l,\ M^+[j] < f,\ M^+[j_1 - 1] \geq f, M^+[j_2 + 1] \geq f\} \leq g$.*

Then we define a consensus motif as follows:

Definition 2.3 *Given a set $\mathcal{S} = \{s_1,\ s_2, \cdots,\ s_n\}$, g a constraint on maximal gap length and f, the minimal frequency required to distinguish between wild-card and specific characters, a consensus motif of length l verifying quorum q for $\mathcal{S}$ is a* pssm $\in I\!R^{+|\Sigma| \times l}$ *such that:*
(1) *M is associated with $\mathcal{S}' \subseteq \mathcal{S} \mid \mathcal{S}' \mid \geq q \times n$,*
(2) *$maxGapLength_{g,f}(M)$ is true.*

3 Sampling sequences for running *pssm*$_s$ to convergence

We deal with n initial sequences of maximal size t. g is the maximal gap length specified. l is the total length of the consensus motif, f is the minimal frequency allowing distinguishing between wild-card and specific characters and ns is the number of specific characters. Both l and ns are *a priori* unknown. In this exposition part, we will not detail which UMA algorithm is used.

Our algorithm implements three levels involving two types of operations on *pssm*$_s$. We describe the algorithm in a bottom up fashion.

3.1 Comparing *pssm*$_s$ at UMA level

The idea underlying our algorithm consists in iterating a local search for *pssm* candidates in sufficiently different contexts. Choosing as such a context a subset of m sequences selected from the initial dataset, a MA is performed on the current sample. Scanning this alignment, we generate at most nm_{MA} *pssm*$_s$, each corresponding to "rectangles" in the MA satisfying the definitions 2.1 and 2.2 at this small scale. Each *pssm* is generated under the following constraints: minimal "height" (number of sequences supporting the *pssm*); minimal "width" (minimal length for candidate motif seeds); the number of contiguous columns with most frequent characters having their frequencies below threshold f is smaller than g (*def.* 2.3); no two *pssm*$_s$ retained among the nm_{MA} ones (at most) have an identical mask (*notation* 5). Two *pssm*$_s$ with the same mask are contracted to yield a unique matrix. The operator designed to contract two *pssm*$_s$ will be described later on. Contracting *pssm*$_s$ is the way chosen to take account of sparse local identical similarities. Studying MA outputs helps understanding that depending on the characteristics of the MA algorithm used and the sample data structure one can not expect that strong local similarities are always well aligned in a "high" rectangle rather than distributed in smaller rectangles. In addition, note that at this low level, two *pssm*$_s$ may have masks such that one is a substring of the other one. Finally, the scoring function used to keep the best nm_{MA} candidates is $fscore_{MA}$, whose choice is discussed further.

3.2 Reducing the search space for motif candidates

As similarities not related to the putative consensus motif searched for are likely to yield strong candidate *pssm*$_s$ (local optimal solutions), we do actually perform a pair of MA_s during each iteration to reject false positives. Each step, a pair of subsets is selected at random for this purpose from the initial data. Once we have obtained $2 \times nm_{MA}$ candidates (at most) for a pair of MA_s, the pairwise comparison of these candidates is processed to identify whether two motif candidates may cooperate to strengthen one of these two motifs. At this level we retain nm_i candidates on the basis of a scoring function $fscore_i$. We mentioned in previous paragraph the

contraction operator. The second operation used at this intermediary level generalizes contraction operation and can be described in words as follows: the *pssm*$_1$ with the greatest number of columns is successively scanned from left to right with a window the size of the other *pssm*$_2$'s second dimension, and yielding each time a potential *pssm*$_{12}$ candidate; this candidate is retained if its score improves the nm_i best current solutions; if so, it replaces *pssm*$_1$. This operation is called *pssm* merging. During a merging operation, wild-card characters are likely to appear, which may entail shortening the resulting candidate motif or may even lead to its rejection for not satisfying the maximal gap length constraint.

Each iteration ends with a pairwise comparison involving the contraction operator. The nm (at most) solutions strenghtened through $i-1$ iterations are compared with the nm_i (at most) solutions just obtained at iteration i. This process yields nm solutions if possible.

The scoring functions retained for candidate filtering during elementary steps ($fscore_{UMA}$), MA confrontations ($fscore_i$) and motif strengthening or rejection through successive iterations ($fscore$) are identical. Contrary to expectation a scoring function rewarding *pssm*$_s$ conserved through iterations (neither successive ones nor scattered ones) was empirically shown to be a bad choice. Furthermore too short candidates must be rejected. The scoring function chosen can not be the usual log-likelihood ratio statistic ($\sum c \in \Sigma \sum_{j=1}^{l} M[c,j] log \frac{M[c,j]}{b[c]}$) where b is the background probability for character c computed on the subset involved in the MA considered. One can only compare *pssm*$_s$ with same second dimension and which are supported by the same number of occurrences. So though locations of occurrences which suppport the best *pssm*$_s$ are memorized through the three levels of the algorithm, we can not use the log-likelihood ratio as a score. Finally our concern is the discovery of consensus motif with *gaps* and we think it is relevant to take account of specific regions only. Gaps are chosen to be "silent" *w.r.t.* the quality of the solution. The scoring function we propose (and test with success) is: $\sum_{j=1,\ M^+[j] \geq f}^{l} {M^+}_f[j]$.

Before we describe formally merging and contraction, we complete the definition of a *pssm* as follows:

Definition 3.1 *A* pssm *is a quadruplet (*pssm *matrix, scoring function f, score, number of sampled sequences supporting this* pssm*). Remember a sequence supports a* pssm *if it contains an occurrence of the consensus motif represented by this* pssm *and contributed to the calculation of its frequencies.*

Definition 3.2
pssm merging: pssm $\times$ pssm $\rightarrow$ pssm *set (eventually $\emptyset$)*
Given pssm$_s$ M_1 *and* M_2 *with characters of* Σ *as line indexes and numbers of columns respectively* l_1 *and* l_2 *($l_1 \leq l_2$ w.l.o.g.), numbers of supporting sequences* ns_1 *and* ns_2 *and* $nm \in \mathbb{N}^+$, $M_1 \oplus M_2$ *is the set of* pssm *matrices* M *of size* $|\Sigma| \times l_1$*, with number of supporting sequences* $ns_1 + ns_2$*, which verifies:*

(1) $\forall c \in \Sigma,\ \forall\ j\ 1 \leq j \leq l_1,\ M[c,j] = \frac{ns_1 M_1[c,j] + ns_2 M_2[c,j_2+j]}{ns_1+ns_2}$*, for some* $j_2 \in [1, l_2 - l_1 + 1]$
(2) $maxGapLength_{g,f}(M)$ *is true.*
(3) $score(M) = \sum_{j=1,\ M^+[j] \geq f}^{l_1} {M^+}_f[j]$
Moreover, let nm *be the maximal number of best candidates retained (*nm *depends on the level considered in the algorithm.)*
(4) $|M1 \oplus M2| \leq nm$ *(Best* pssm$_s$ *are kept considering the highest scores.)*

Definition 3.3
pssm contraction: pssm $\times$ pssm $\rightarrow$ pssm *(eventually null)*
Given pssm$_s$ M_1 *and* M_2 *having the same number of columns* l*,* $M1 \bullet M2$ *is the* pssm *verifying*
(1) $M_1 \bullet M_2 \in M_1 \otimes M_2$
(2) *mask(M_1) = mask(M_2).*

One must be aware that *pssm* merging does not necessarily correspond to mask intersection. That is two *pssm*$_s$ with respective specific characters c_1 and c_2 in position j may yield a merged $pssm$ with a specific character differing from the formers in this position.

As already mentioned locations of occurrences suppporting the best *pssm*$_s$ are memorized through all three levels of the algorithm. As candidate motifs are strengthened and their *pssm*$_s$ are merged, the occurrence locations relative to these *pssm*$_s$ are strengthened too ("one more hit"). As we had foreseen it starting this research work, the optimal solution may well be splitted in sub-motifs among the nm final sub-optimal solutions, either overlapping

or not. There are two reasons for this: depending on the MA algorithm run, large gaps may induce splitted local similarity detection; when the motif length specified as a input for the UMA is smaller than the real unknown consensus motif length. So the ending stage of our algorithm checks for co-occurrence between as many sub-motifs as possible, under the quorum constraint. Last step consists in checking the significance of the motif after its retrieval.

3.3 Checking the motif significance

Such consensus motifs involved in biological functions contain intrinsic information. The probability that a given motif of length l with ns specific characters occurs by chance at least one time in each of n sequences of size t is:$(1-(1-\sum_{m=0}^{d}\binom{ns}{m}(\frac{|\Sigma|-1}{|\Sigma|})^{m}(\frac{1}{|\Sigma|})^{ns-m})^{t-l+1})^{n}$. The principle of our algorithm is to retrieve occurrences *w.r.t.* the minimal frequency for specific characters. So the maximal Hamming distance observed for occurrences must be calculated *a posteriori*. It has been shown (see [13] for example) that this value, which is calculated with no care of overlapping words is though a good approximation for the theoretical value. The principle of our algorithm is to retrieve occurrences *w.r.t.* the minimal frequency for specific characters. So the maximal Hamming distance observed for occurrences must be calculated *a posteriori*.

The algorithm is described in 1.

3.4 Complexity

In average, every sequence will be chosen $\frac{mu}{n}$ times during u iterations since all samples are equiprobable. The maximal complexity for aligning z sequences of maximal length t with ClustalW is $O(z^3t^2)$ (steps 8 and 9). Thus, with u iterations each performing a pair of multiple alignments, the computation cost is $O(u\ t^2m^3)$. Scanning the MA generated from a small sample of sequences to identify nm_{MA} $pssm_s$ has a negligible cost. Finally, the complexity is $O(u\,t^2\,C)$ with C constant if m is chosen small, which is the case in our implementation (10). When using MoDEL, the time complexity of an alignment is approximated by $O(n\ t\ w\ b)$ where w is the fixed length for the local MA (the range $[15-25]$ for binding site retrieval) and b is the intrinsic number of iterations for MoDEL execution (default value is 45). The complexity is then $O(u\ m\ t\ w\ b)$. Note that the time complexity does not depend on the maximal gap length g.

4 Results

4.1 First benchmark: simulated data

We implemented supplementary software devoted to the algorithm evaluation. Depending on the aim, one specifies the re-use of a given consensus motif or one creates a new mask: the software designed for this purpose requires the length of the consensus motif, the number of gaps, the range for gap lengths. Artificial sequences of sizes in a specified range are then generated at random and occurrences of a given consensus motif are inserted under (strict) quorum constraint. Then the generation algorithm blurs all motif occurrences at random, but under the constraint of the minimal frequency required for specific characters.

Evaluation is done as follows: comparing the retrieved consensus motif m with the real one M, we systematically compute two parameters named $cover$ and $exactness$. $cover$ is the ratio $|m|\ /\ |M|$ where $|m|$ denotes the length of m. *W.l.o.g.* suppose $|m|\ <\ |M|$. For any valid position in M, we calculate the mismatch score between m and the current sub-word in M having length $|m|$ (mismatch score between any two different characters $= 1$), skipping the z (for instance) wild-card characters in M. $exactness$ is the maximum value among ratios $\frac{|m|-z-mismatch\ score}{|m|-z}$.

Our first tests were run using the MA software clustalW (v. 1.83) [15]. We tuned it to allow long shifts of the sequences relatively to one another so as to only have gaps (those of clustalW) at extremities. We chose the identity matrix to set substitution costs. We focused on 9 consensus motifs to test the case ($f = 100\%, quorum = 100\%$). These motifs are presented in Table 1. We observed the cover was not perfect but that we could yet recover rather fuzzy motifs (motifs (6), (8) and (9)). Introducing motif extension (see 3.2), we obtained a quasi perfect cover except for motif (9) which is TxxxCTTxxxCxxxT. Table 2 shows this improvement.

Algorithm 1 Gapped consensus motif retrieval

Input: a set S of n sequences; maximal gap length g; minimal frequency f; quorum q; sample size m.
Output: answer Yes/No; if Yes, a mask $\mathcal{M}ask$, the corresponding *pssm* P, occurrence locations $OccLoc$ for at least $q \times n$ sequences.

while no solution is found **do**
  $\mathcal{S}ol \leftarrow \emptyset$ /* Initialize the list of solutions ($pssm_s$).*/
  for iter=1, $\cdots$, u **do**
    $S_1 \leftarrow AleatSubset(S, m)$ /* first sample */
    $S_2 \leftarrow AleatSubset(S, m)$ /* second sample */
    /* Now perform 2 *ungapped* multiple alignments to obtain $2 \times nm_{UMA}$ best solutions $\mathcal{S}ol_1 \cup \mathcal{S}ol_2$.*/
    $\mathcal{S}ol_1 \leftarrow UMA(S_1, pssmContractionOperator, fscore_{UMA})$
    $\mathcal{S}ol_2 \leftarrow UMA(S_2, pssmContractionOperator, fscore_{UMA})$
    /* Compare and keep nm_i best solutions $\mathcal{S}ol_i$.*/
    $\mathcal{S}ol_i \leftarrow pairwiseComparison(pssmMergingOperator, \mathcal{S}ol_1, \mathcal{S}ol_2, fscore_i)$
    /* Update the nm best solutions in $\mathcal{S}ol$.*/
    $\mathcal{S}ol \leftarrow pairwiseComparison(pssmMergingOperator, \mathcal{S}ol, \mathcal{S}ol_i, fscore)$
  end for
  if $\mathcal{S}ol \neq \emptyset$ **then**
    $answer \leftarrow Yes$
    /* Check for co-occurrence between some masks in $\mathcal{S}$ and subsequently extend the consensus motif (if necessary) until the quorum constraint is no more satisfied. Doing this, reject occurrence locations which are not consistent with quorum constraint. Then check for the significance of the motif (see text 3.3).*/
    $(\mathcal{M}ask, P, occLoc) \leftarrow extendMotifSelectOccurrences(\mathcal{S}ol, S, q)$
    break;
  end if
end while
$answer \leftarrow No$

consensus motif number	nb gaps	length of gaps	consensus motif
1	1	1	CTGATCGxTGACTAT
2		2	GTTxxGGTTTTAAGT
3		3	TCCACGxxxTTGGTC
4	2	1	CTAGxGAAxATTTAT
5		2	GCGxxATxxAACTGC
6		3	AGTxxxCxxxCCCAA
7	3	1	GTxGTTGTxAGxAGC
8		2	CAxxCGxxTCxxGCT
9		3	TxxxCTTxxxCxxxT

Table 1: Nine consensus motifs of length 15.

consensus motif number	# gaps	gap length	cover	(a) exactness	time (s)	cover	(b) exactness	time	motif significance see text 3.3
1	1	1	0.86	1.0	347	0.96	0.99	207	0.998
2		2	0.77	1.0	431	0.95	0.99	105	0.016
3		3	0.96	1.0	493	1.19	0.99	577	2.812 E-15
4	2	1	0.85	1.0	512	0.92	1.0	195	0.016
5		2	0.67	1.0	425	0.96	0.93	97	5.584 E-36
6		3	0.97	0.99	465	0.96	0.99	87	4.825 E-83
7	3	1	0.79	1.0	367	0.94	0.92	95	2.812 E-15
8		2	0.71	1.0	409	0.95	0.99	83	4.825 E-83
9		3	0.71	0.95	426	0.65	0.98	105	3.231 E-155

Table 2: Tests for the 9 motifs of Table 1 in the exact case (minimal frequency = 100%). Average values for cover, exactness and execution time are computed from 100 runs over the same set of 40 sequences of length 50, for each consensus motif. For purpose comparison, the number of iterations of the algorithm is set to 100. Version (b) of the algorithm implements motif extension. The right column displays the probability that the *exact* motif occurs by chance at least once in the set of 40 sequences (see text 3.3).

Then we opted for the MA software MoDEL [6] to evaluate our method under a larger variety of conditions: longer sequences, various gap distributions and various frequencies for specific characters. Besides statistics about $cover$ and $exactness$, Table 3 shows a series of results including more details on false wild-card or specific characters predicted. Table 4 gives the significances (3.3) of the motifs considered depending on different mismatch values. The reader familiar with motif discovery topics will have noticed in the previous benchmark peculiar instances of the so-called challenge problem [14] ($b,\ c,\ d$ and e) . The challenge for instance (n, t, l, d) of this problem consists in discovering in n sequences of size t an occurrence of a motif of length l with exactly d mismatches. In our case, difficulty is increased by the presence of gaps. Furthermore an instance is indeed (n, t, ns, d) with ns the number of specific characters and d the *maximal* mismatch value. Remember we generate data controlling the mismatch errors by column, and not by occurrence.

	a1	a2	a3	a4	b1	b2	c1	c2	d1	d2	e1	e2
q %	100	70	100	70	100	70	100	70	100	70	100	70
f %	100	100	80	80	80	80	80	80	80	80	80	80
cover	0.97	0.97	0.85	0.68	1.0	1.0	1.0	0.97	0.92	0.93	0.91	0.95
exactness	1.0	1.0	0.98	0.92	0.98	0.96	0.95	0.92	0.96	0.95	1.0	1.0
false w.	0.49	0.5	2	2.4	0.5	1.43	0.2	1.02	0.37	0.47	0.11	0.3
false s.	0.38	0.37	0.13	0.9	0	0.44	0.9	0.3	0.22	0.21	0.2	1.5

motif		ns
a	GCGxxAAGCAxxCC	10
b	TGATxxTAGxxACGCC	12
c	TTTxCTCxCGxCCGxGAG	14
d	AATTTxxTCCTAGxTxTACG	16
e	CTTGGACxCGAxCCTCxxCGCC	18

Table 3: Performances of the algorithm for 2 quorum values (q) and 2 frequency values (f). In each subcase a 100 sequences of lengths ranging from 50 to 300 have been generated under the quorum and minimal frequency constraints. In cases $b,\ c,\ d$ and e, 20 sequences of length 600 have been generated. Average values for cover, exactness and number of false wild-card (resp. specific) characters predicted have been computed for 100 runs (model length specified for the MA software MoDEL: 20). ns denotes the number of specific characters in the motif.

motif	n	t	l	ns	d	significance of the motif
a	100	50	14	10	2	2.422 E-182
				4	5.374 E-29	
				7	0.999	
a	100	300	14	10	2	1.318 E-95
				4	0.720	
				7	1.0	
b	20	600	16	12	3	1.686 E-14
c			18	14	4	1.383 E-15
d			20	16	5	4.776 E-17
e			22	18	6	9.0912 E-19

Table 4: Significance of various classes of consensus motifs from Table 3 (see text 3.3). n is the number of sequences, t is their common length, l is the motif length, ns is the number of specific characters, d is the maximal number of mismatches per occurrence (computed on specific characters only).

4.2 Second benchmark: biological data

Then we achieved a more thorough examination of real *vs* predicted occurrence locations focusing on a hard case: long blurred occurrences. We wanted to focus on motifs of a biological type. Yet we still wanted to control the distance of the occurrences *w.r.t.* the motifs. Inserting in 50 artificial sequences (quorum 100%) the tandem binding site described in [12] (AxTGAATAAxxATxCATxTATxxTGAATAxAAATTCAxT) as the consensus of 8 transcription promoters for ArgR in *E. Coli*, we *imposed* the specific character frequency 70% and *observed* rather blurred occurrences (up to 13 mismatches for the 30 specific characters, average number of mismatches 9). In this case the UMA performed on the whole dataset yields a percentage of false locations equal to 38% in average (10 runs). We always managed to obtain the good locations (among other ones, since our current version does not choose between candidate locations) over 10 runs. The cover parameter averages 76%, which shows motif extension completed the search in this case since MoDEL was run for the model length 20. The 10 runs yielded 5.3 false specific characters in average and 1.2 false wild-cards. Table 5 gives the 3 motifs we obtained together with the motif discovered by MEME. Table 6 gives the motif significance for different mismatch values.

	# results over 10 runs	# false specific characters	# false wild-card characters
AxTGAATAAxxATxCATxTATxxTGAATAxAAATTCAxT (consensus motif)			
TGAATAATAATACATxTATTGTGAATAAA	3	6	0
TGAATAATAxTACATxTATTxTGxATAAAA	5	5	2
TGAAxAATAATACATxTAxAxTxAATxAAA	2	5	4
MEME result AxTGAATAAxxATxCATxTATxxTGAATAxAAATTCAxT (1)			

Table 5: The 3 motifs retrieved in 10 runs for 50 artificial sequences of lengths ranging from 50 to 300, all containing an occurrence of the tandem binding site AxTGAATAAxxATxCATxTATxxTGAATAxAAATTCAxT (consensus of 8 transcription promoters for ArgR in *E. Coli* collected in [12]). (1) Specific characters were identified from the MEME output as contributing for a value greater than or equal to 0.6 bits to total information content IC (19.5 bits).

n	t	l	ns	d	significance of the motif
50	50	39	30	7	0.0
				11	2.618 E-196
				15	3.928 E-75
50	300	39	30	7	2.708 E-298
				11	2.220 E-129
				15	3.504 E-15

Table 6: Significance (see text 3.3) of the consensus motif of Table 5 for different mismatch values (d) and different sequence sizes (t). n is the number of sequences, t is the sequence length, l is the motif length, ns is the number of specific characters, d is the maximal number of mismatches per occurrence (computed on specific characters only).

Finally, we run our algorithm on the same biological datasets as those collected for STARS evaluation from

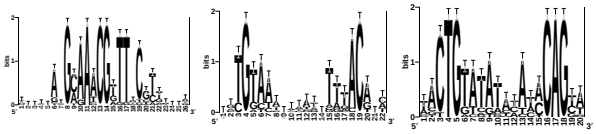


Figure 2: (a) Sequence logos for binding sites of genes from *E.Coli* coding for different proteins. From left to right: 18 genes coding for PurR protein - 5 genes coding for TyrR protein - 16 genes coding for LexA protein. All datasets were collected by A.Mancheron [12] who computed the corresponding sequence logos.

the pages *http://www.softberry.com/berry.phtml* and *http://arep.med.harvard.edu/ecoli_matrices/*. Figure 2 shows the sequence logos collected in [12]. Table 7 describes the conditions for the tests and their results. We compared our results with the outputs of MEME and STARS.

	Tata box for Dicot plants	Binding sites PurR, E.Coli	Binding sites TyrR, E.Coli	Binding sites LexA, E.Coli
# seq.	131	18	5	16
length	251-251	299-5864	251-2021	100-3842
avg length	251	2477	717	1809
# nucleot.	32880	44592	3585	28941
motif (1)	TATAxATA	AxGxAAxCGxTTxCxT	TGTAAxxxxxxxTTxAC	CTGTAxAxxxAxxCAG
our algor.				
q a 100%	TATAAATA	GxAAxCGxTTxC AxGxAAACGTTTxCGT	TGTAxxTTTxTxxxTACA	TACTGTATATxxAxxCAG
q b 70% (2)	ATAAAxA TAxAAA(3)	GxAAACGTTTxCG GCAAACGTxTxCxT		CTGTxxATCxATACAG CTGTxxATxxATACAGTA
MEME				
q 100%	TATAAATA	AxGCAAACGxTTxCxT (5)	TTxTxTxTTAACCxxCTxCCC (7) TTTTAxGxxxCTGCCCGTxxxCxT (8)	CTGTATATxxAxxCAG (9)
q 70%	TATAAATA	AxGxAAACGTTTxCxT (6)		CTGxAxAxxxAxACAGxA (10)
STARS (4)	TATAAATA	GCxAxCGTTTTC	TGTAAAxxAAxxTxTAC	ACTGTATATxxAxxCAG

Table 7: Outputs of our algorithm, MEME and STARS for biological motif retrieval under minimal frequency constraint $f = 80\%$ and 2 quorum values. (1) The biological consensus motifs are described with the sequence logos shown in Figure 2. We systematically compared 10 outputs for our algorithm. (2) To test our method with quorum q, we replaced $100 - q\%$ sequences in the initial dataset with as many sequences of the same lengths chosen at random in the adequate genomes. (3) The 2 sub-optimal solutions found for quorum $< 100\%$. (4) q = 100%, published results. We obtained *gapped* predicted consensi from MEME outputs under the following conditions: (5) contribution to information content IC (20.4 bits) strictly below 0.8 bits for specific characters. (6) 1.0 bits and total IC 18.8 bits - see (4). (7) when length specified in [15,25]; threshold 0.9 bits and total IC 25.3 bits. (8) when length specified in [20,25]; threshold 0.9 bits and total IC 28.0 bits. (9) when length specified = 20; threshold 1.0 bits and total IC 23.1 bits. (10) when length specified = 20; threshold 1.0 bits and total IC 18.8 bits. [12].

4.3 Discussion

Not only is our algorithm efficient on sets of 100 sequences with lengths ranging from 50 to 300. It succeeds too in recovering the consensus motif for sequences with lengths up to 5800 and averaging 2500. We designed the 4 motifs b, c, d and e (Table 3) to confront our algorithm to the challenge problem mentioned above (a class of subtle motifs, see end of 4.1). We manage to recover such motifs, though in our case they are gapped, which brings more difficulty. Our stochastic non sophisticated approach yields relatively steady results. The quality of the outputs (both in terms of motif and occurrence recovery) stands comparison with MEME and STARS. All three methods perform well in motif identification, except for MEME on the TyrR set (5 sequences, gap length:7). Overmore, we checked that the real occurrences were always found among the occurrences most frequently hit during u iterations of our algorithm. Resorting to a much less sophisticated technique than the one implemented in MEME, our algorithm compares favourably with MEME and STARS in terms of efficiency. MEME yielded outputs in less than 30 s. Our prime objective being prooving the soundness and accuracy of our method, we just ran our algorithm on the usual PC type (RAM 256 Mo, 1.6 GHz), with no iteration number optimization suited to the data analysed ($u = 100$). Finally, accuracy holds for quorum under 100%.

5 Conclusion

We presented a novel method for $UCMR$ under minimal specific character frequency, maximal gap length and quorum constraints. First, we wanted to benefit from a pre-existing local multiple alignment algorithm for our purpose. Secondly, *pssm* convergence is obtained in an original way: strengthening (literally *merging*) the best candidates satisfying frequency and gap constraints, which distinguishes our algorithm from methods such as MEME and PROJECTIONS. Furthermore, our algorithm checks for the maximal gap length constraint during all three levels, instead of performing *a posteriori* checking. As for other stochastic methods, prooving the soundness of the intuition behind the method required implementing an experimental protocol. Our algorithm is efficient and robust *w.r.t.* the following criterion: maximal gap length specified with too high a value. Retrieval under quorum constraint is successful, which is not a trivial result. Moreover, our method has a low memory cost. Next step will consist in examining which parts of the algorithm may significantly be sped up. Studying the convergence of the solution with respect to the number of iterations is also one of our future tasks as will be a more thorough examination of the choice for scoring functions. These former topics are currently in study. Finally, testing the algorithm on other more various biological inputs and systematically comparing the locations predicted by our method and other approaches will be of high interest.

Acknowledgements

We wish to thank Alban Mancheron for kindly putting at our disposal datasets he collected and the corresponding sequence logos he computed. Thanks is also due to David Hernandez for providing the beta version of the MODEL software. This material is based upon work supported by the French National Center for Scientific Research (C.N.R.S.).

References

[1] Akutsu, T., Arimura, H., Shimozono, S.: On approximation algorithms for local multiple alignment. In: Proceedings of the 4th Annual International Conference on Computational Molecular Biology, Tokyo. ACM Press, 1-17. (2000)

[2] Altschul, S., Gish, W., Miller, W., Myers, E., Lipman, D.: Basic Local Alignment Search Tool. Journal of Molecular Biology, 215, 403-410. (1990)

[3] Bailey, T.L.: Likelihood vs. information in aligning biopolymer sequences. USCD technical report cs93-318. University of California, San Diego. (1993)

[4] Bailey, T.L., Elkan, C.: Unsupervised learning of multiple motifs in biopolymers using expectation maximization. Machine learning, 21, 51-80. (1995)

[5] Buhler, J., Tompa, M: Finding motifs using random projections, in Proceedings of the Fifth International Conference on Computational Molecular Biology (RECOMB), 69-76, Montréal, Canada, ACM Press, apr. (2001)

[6] Hernandez, D., Gras, R., Appel, R.: MODEL: an efficient strategy for ungapped local multiple alignment. Computational Biology and Chemistry, 28, 2, 119-128, apr. (2004)

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

[8] Kulback, S.: Information theory and statistics. Dover publications, New York. (1968)

[9] Lawrence, C.E., Altschul, S.F., Boguski, M.S., Liu, J.S., Neuwald A.F., Wootton J.C.: Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. Science, 262, 208–214, oct. (1993)

[10] Lipman, D., Pearson, W.: Improved tools for biological sequence comparison. Proc. Nat. Acad. Sci. USA, 85, 2444-2448. (1988)

[11] Ma, B., Tromp, J., Li, M.: Pattern Hunter: faster and more sensitive homology search. Bioinformatics, 18, 3, 440-445. (2002)

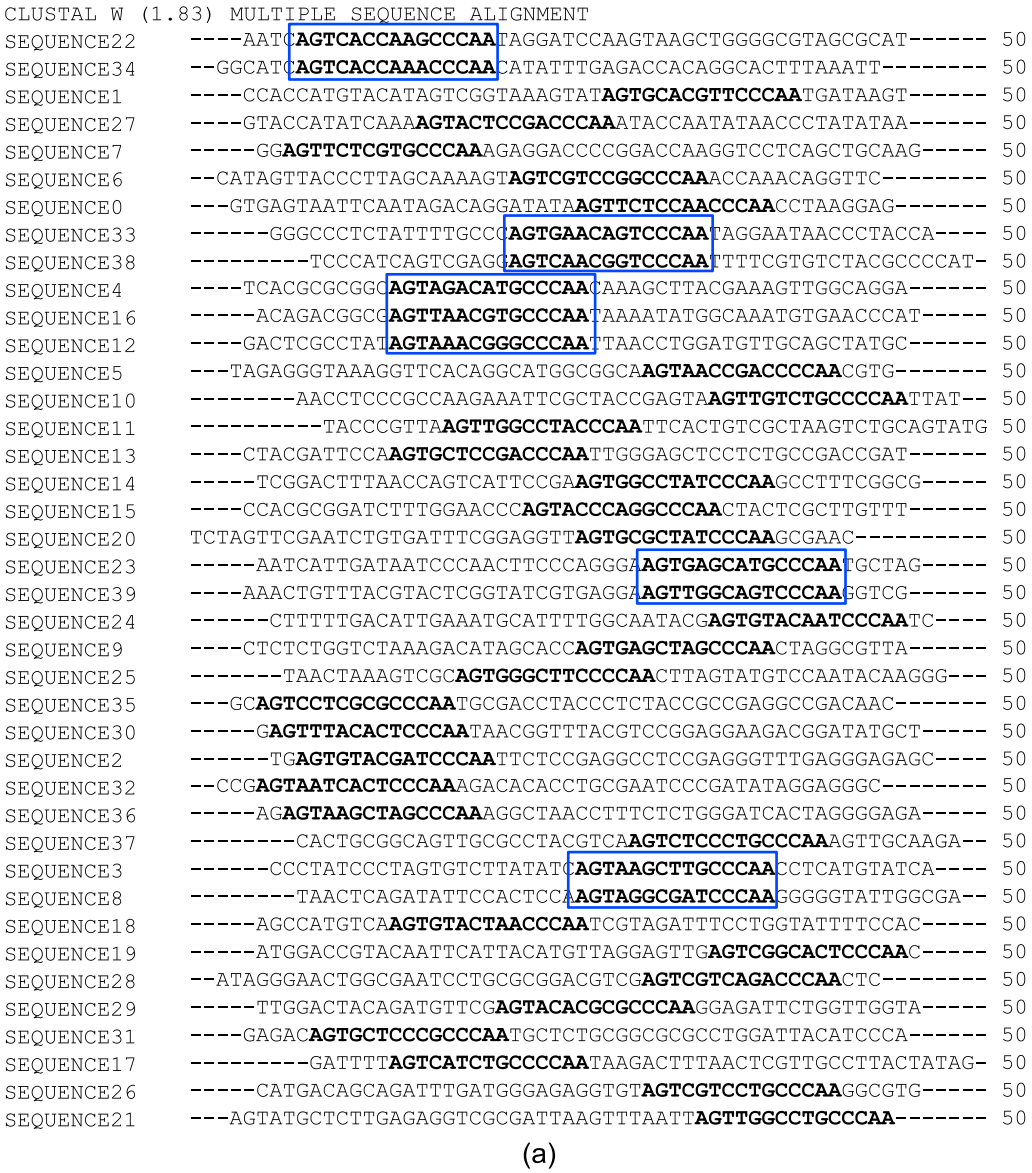
[12] Mancheron, A., Rusu, I.: Pattern discovery allowing gaps, substitution matrices and multiple score functions. Proceedings of the Third Workshop of Algorithms in Bioinformatics WABI, 2812, 129–145, Budapest, Hungary, Springer-Verlag, LNBI, sep. (2003)

[13] Nicodème, P.: q-gram analysis and urn models Proceedings of Discrete Random Walks DRW2003, 243-258. (2003)

[14] Pevzner, P.A., Sze, S.-H.: Combinatorial algorithm for finding subtle signals in DNA sequences. Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology, 269-278, San Diego, California, aug. (2000)

[15] Thompson, J.D., Higgins, D.G., Gibson T.J.: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. Nucleic Acids Research, 22, 4673-4680. (1994)

Appendix



(a)

Figure 1: Multiple alignment obtained with software CLUSTALW for a set of 40 sequences all containing the consensus motif $AGTxxxCxxxCCCAA$.

LABORATOIRE D'INFORMATIQUE DE NANTES-ATLANTIQUE

Gapped consensus motif discovery: evaluation of a new algorithm based on local multiple alignments and a sampling strategy

Christine Sinoquet

Abstract

We check the efficiency and faisability of a novel method designed for the discovery of *a priori* unknown motifs described as gaps alternating with specific regions. Such motifs are searched for as consensi of **non homologous** biological sequences. The only specifications required concern the maximal gap length, the minimal frequency for specific characters and the minimal percentage (quorum) of sequences sharing the motif. Our method is based on a cooperation between a multiple alignment method for a quick detection of local similarities and a sampling strategy running candidate position specific scoring matrices to convergence. This rather original way implemented for converging to the solution proves efficient both on simulated data, gapped instances of the so-called challenge problem, promoter sites in Dicot plants and transcription factor binding sites in *E.Coli*. Our algorithm compares favorably with the MEME and STARS approaches in terms of accuracy.

LINA, Université de Nantes
2, rue de la Houssinière
B.P. 92208 — F-44322 NANTES CEDEX 3