JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 18, Number 4, 2011

Pp. 535–545
DOI: 10.1089/cmb.2010.0177

Research Articles

# MADMX: A Strategy for Maximal Dense Motif Extraction

ROBERTO GROSSI,[1] ANDREA PIETRACAPRINA,[2] NADIA PISANTI,[1]
GEPPINO PUCCI,[2] ELI UPFAL,[3] and FABIO VANDIN[3]

## ABSTRACT

**We develop, analyze, and experiment with a new tool, called MADMX, which extracts frequent motifs from biological sequences. We introduce the notion of *density* to single out the "significant" motifs. The density is a simple and flexible measure for bounding the number of don't cares in a motif, defined as the fraction of solid (i.e., different from don't care) characters in the motif. A *maximal dense motif* has density above a certain threshold, and any further specialization of a don't care symbol in it or any extension of its boundaries decreases its number of occurrences in the input sequence. By extracting *only* maximal dense motifs, MADMX reduces the output size and improves performance, while enhancing the quality of the discoveries. The efficiency of our approach relies on a newly defined combining operation, dubbed *fusion*, which allows for the construction of maximal dense motifs in a bottom-up fashion, while avoiding the generation of nonmaximal ones. We provide experimental evidence of the efficiency and the quality of the motifs returned by MADMX.**

**Key words:** algorithms, motifs extraction.

## 1. INTRODUCTION

The discovery of frequent patterns (*motifs*) in biological sequences has attracted wide interest in recent years, due to the understanding that sequence similarity is often a necessary condition for functional correlation. Among other applications, motif discovery is an important tool for identifying regulatory regions and binding sites in the study of functional genomics. From a computational point of view, a major complication for the discovery of motifs is that they may feature some sequence variation without loss of functionalities. The discovery process must therefore target *approximate motifs*, whose occurrences are similar but not necessarily identical. Approximate motifs are often modeled through the use of the *don't care* character in certain positions, which is a wild card matching all characters of the alphabet, called *solid characters* (Parida, 2008).

Finding interesting approximate motifs is computationally challenging. The output may explode combinatorially for an increasing number of don't cares and/or a decreasing value of the minimum frequency threshold. This explosion is not mitigated if the discovery targets are just the maximal motifs—a subset of

[1]Dipartimento di Informatica, Università di Pisa, Italy.
[2]Dipartimento di Ingegneria dell'Informazione, Università di Padova, Italy.
[3]Department of Computer Science, Brown University, Providence, Rhode Island.

the motifs which implicitly represents the full set. Even if the final output is not too large, partial data during the intermediate steps might lead to memory saturation or to extensive computation.

In this paper, we focus on the discovery of approximate motifs, which contain blocks of solid characters (solid blocks) separated by one or more don't cares. We propose a general approach for controlling the number of don't cares in these motifs. Specifically, we introduce the notion of *dense motif*, a motif where the *fraction* of solid characters is above a given threshold. Our density notion is flexible and general, since it allows for arbitrarily long runs of don't cares as long as the fraction of solid characters in the pattern is above the threshold. The rationale is that sparse motifs are less interesting in biological sequences, unlike what happens to frequent itemsets in baskets (where any subsets of correlated items are of interest). We define a natural notion of *maximality* for dense patterns and devise an efficient algorithm, called MADMX (pronounced *Mad Max*), which performs complete MAximal Dense Motif eXtraction from an input sequence, with respect to user-specified frequency and density thresholds.

The key technical result at the core of our extraction strategy is a closure property which affords the complete generation of all maximal dense motifs in a breadth-first fashion, through an *apriori*-like strategy (Agrawal and Srikant, 1994). It starts from a relatively small set of solid blocks, and then repeatedly applies a suitable combining operator, called *fusion*, to pairs of previously generated motifs. In this fashion, our strategy avoids the generation and consequent storage of intermediate patterns which are *not* in the output set, which ensures time and space complexities polynomial in the combined size of the input and the output.

The problem of motif discovery and its variants have been addressed by a large body of literature in the last decade (Parida, 2000; Apostolico and Parida, 2004; Pisanti et al., 2005; Apostolico and Tagliacollo, 2007, 2009; Ukkonen, 2007; Morris et al., 2008; Arimura and Uno, 2008; Apostolico et al., 2009), and an excellent survey of known results can be found in Parida (2008). In order to alleviate the computational burden of motif extraction and to limit the output to the most promising or interesting discoveries, some works combine the traditional use of a frequency threshold with restrictions on the flexibility of the extracted motifs, often captured by limitations on the number of occurring don't cares.

In a recent work, Apostolico et al. (2009) studied the extraction of *extensible motifs*, comprising standard don't cares and extensible wild cards. The latter are spacers of variable length that can take different size (within pre-specified limits) in each occurrence of the motif. An efficient tool, called VARUN, is devised in Apostolico et al. (2009) for extracting all maximal extensible motifs (according to a suitable notion of maximality defined in the article) which occur with frequency above a given threshold $\sigma$ and with upper limits $D$ on the length of the spacers. VARUN returns the extracted motifs sorted by decreasing z-score, a widely adopted statistical measure of interestingness. The authors demonstrate the effectiveness of their approach both theoretically, by proving that each maximal motif features the highest z-score within the class of motifs it represents, and experimentally, by showing that the returned top-scored motifs comprise biologically relevant ones when run on protein families and DNA sequences. It has to be remarked that the motifs studied in our work are *rigid*, in the sense that extensible wild cards are not allowed.

An alternative way of limiting the number of don't cares in a motif has been explored in Rigoutsos and Floratos (1998). The authors define $\langle L,W\rangle$ motifs, for $L \leq W$, where at least $L$ solid characters must occur in each substring of length $W$ of the motif. They propose a strategy for extracting $\langle L,W\rangle$ motifs which are also maximal, although their notion of maximality is not internal to the class of $\langle L,W\rangle$ motifs. As a consequence, the algorithm is not complete, since it disregards all those $\langle L,W\rangle$ motifs that are not subsumed by a maximal $\langle L,W\rangle$ one.

We performed a number of experiments on MADMX to assess the biological significance of maximal dense motifs and to compare MADMX against its most recent and close competitor VARUN. For the first objective, we used MADMX to extract maximal dense motifs from a number of human DNA fragments. We compared the output set against those in `RepBase` (Jurka et al., 2005), the largest repository of repetitive patterns for eukaryotic species, using REPEATMASKER (Smit et al., 1996), a popular tool for masking repetitive DNA. The experiments show that all of our returned motifs are occurrences of patterns in RepBase, and *fully* characterize the family of SINE/ALU repeats (and partially the LINE/L1 family). This provides evidence that the notion of density, when applied to rigid motifs, captures biological significance.

Next we compared the z-score performance of MADMX and VARUN. We ran both algorithms on several families of DNA fragments, limiting VARUN to the generation of rigid motifs and setting the parameters so as to obtain comparable output sizes, with motifs listed by decreasing z-score. The experiments show that the top-$m$ highest-ranking motifs returned by MADMX almost always feature higher z-scores than the corresponding top-$m$ ones returned by VARUN, even for large values of $m$, with only a modest increase in running time, which may be partly due to the fact that coding of MADMX is yet to be optimized.

This article is organized as follows. In Section 2, several technical definitions and properties of motifs with don't cares are given. Section 3 proves the closure property at the base of MADMX and provides a high-level description of the algorithm. In Section 4, the experimental validation of MADMX is presented.

## 2. PRELIMINARY DEFINITIONS AND PROPERTIES

Let $\Sigma$ be an alphabet of $m$ characters and let $s = s[0]s[1]\dots s[n-1]$ be a string of length $n$ over $\Sigma$. We use $s[i\dots j]$ to denote the substring $s[i]s[i+1]\cdots s[j]$ of $s$, for $i \le j$. Characters in $\Sigma$ are also called *solid characters*. We use $\circ \notin \Sigma$ to denote a distinguished character called *wild card* or *don't care* character. Let $\varepsilon$ denote the empty string. A *pattern* $x$ is a string in $\{\varepsilon\} \cup \Sigma \cup \Sigma(\Sigma \cup \{\circ\})^*\Sigma$. However, whenever necessary, we will assume that patterns are implicitly padded to their left and right with arbitrary sequences of don't care characters.

Given two patterns $x$, $y$ we say that $y$ is *more specific* than $x$, and write $x \preceq y$, iff for every $i \ge 0$ either $x[i] = y[i]$ or $x[i] = \circ$. If $x \preceq y$ and $x \ne y$, we write that $x \prec y$. Given two patterns $x$, $y$, we say that $x$ *occurs in* $y$ *at position* $\ell$ or, alternatively, that $y$ *contains* $x$ iff $x \preceq y[\ell \dots \ell + |x| - 1]$. If $x \prec y$ we say that $y$ *strictly contains* $x$. For a string $s$, the *location list* $\mathcal{L}_x$ of a pattern $x$ in $s$ is the complete set of positions at which $x$ occurs in $s$. We refer to $f(x) = |\mathcal{L}_x|$ as the *frequency* of pattern $x$ in $s$. (Note that $f(\varepsilon) = n$.) As in Ukkonen (2007), the *translated representation* of the location list $\mathcal{L}_x = \{l_0, l_1, l_2, \dots, l_k\}$ is $\tau(\mathcal{L}_x) = \{l_1 - l_0, l_2 - l_0, \dots, l_k - l_0\}$. Given two patterns $x$, $y$, we say that $y$ *subsumes* $x$ in $s$ if $f(x) = f(y)$ and $y$ contains $x$. As a consequence, $y$ subsumes $x$ if and only if $\tau(\mathcal{L}_x) = \tau(\mathcal{L}_y)$. A pattern $x$ is *maximal* if it is not subsumed by any other pattern $y$. (We observe that this notion of maximality coincides with that of Pisanti et al., 2005.) Given a pattern $x$, its *maximal extension* $\mathcal{M}(x)$ is the maximal pattern that subsumes $x$, which can be shown to be unique (Pisanti et al., 2005).

In what follows, we call *solid block* a string in $\Sigma^+$ and a *don't care block* a string in $\{\circ\}^+$. Furthermore, given a pattern $x$, the number of don't care characters contained in $x$ is denoted by $dc(x)$.

**Definition 1.** *The* density $\delta(x)$ *of* $x$ *is:* $\delta(x) = 1 - dc(x)/|x|$. *Given a (density) threshold* $\rho$, $0 < \rho \le 1$, *we say that a pattern* $x$ *is dense if* $\delta(x) \ge \rho$.

Note that a solid block is a dense pattern with respect to every threshold $\rho$.

We concentrate our attention on dense patterns that are not subsumed by any other dense pattern: they are the most interesting dense representatives in the equivalence classes induced by the relation that puts any two *dense* patterns $x$ and $y$ together if and only if $\tau(\mathcal{L}_x) = \tau(\mathcal{L}_y)$, as defined below.

**Definition 2.** *A dense pattern* $x$ *is a* maximal dense pattern *in* $s$ *if it is not subsumed by any dense pattern* $x' \ne x$.

Observe that a maximal dense pattern $x$ needs not be a maximal pattern in the general sense, since $\mathcal{M}(x)$ might be a nondense pattern. However, every dense pattern $x$ is subsumed by *at least* one maximal dense pattern. In fact, it is easy to see that all of the maximal dense patterns that subsume $x$ are dense substrings of $\mathcal{M}(x)$, namely, those that contain $x$ and are not substrings of any other dense substring of $\mathcal{M}(x)$. We want to stress that there might be several maximal dense patterns that subsume $x$. As an example, for $\rho = 2/3$, the dense pattern $x = \mathrm{B}$ in the string $S = \texttt{AdBeCfAgBhC}$ is subsumed by maximal dense patterns $\mathrm{A} \circ \mathrm{B}$ and $\mathrm{B} \circ \mathrm{C}$, while $\mathcal{M}(x) = \mathrm{A} \circ \mathrm{B} \circ \mathrm{C}$ is not dense.

**Definition 3.** *Given a frequency threshold* $\sigma$ *and a density threshold* $\rho$, *a pattern* $x$ *is a* dense maximal motif *in* $s$ *if* $x$ *is a maximal dense pattern in* $s$ *with respect to* $\rho$, *and* $f(x) \ge \sigma$. *A dense maximal motif for* $\rho = 1$ *is also referred to as* maximal solid block.

**Problem of interest.** We are given an input string $s$, a frequency threshold $\sigma$, and a density threshold $\rho$. Find all the maximal dense motifs in $s$.

In the example above, the maximal dense motifs for $\sigma = 2$, $\rho = 2/3$ are $\mathrm{A} \circ \mathrm{B}$ and $\mathrm{B} \circ \mathrm{C}$. In the rest of the article, we will omit referencing the input string $s$ when clear from the context.

## 3. AN ALGORITHM FOR MAXIMAL DENSE MOTIF EXTRACTION

In this section, we describe our algorithm, called MADMX for MAximal Dense Motif eXtraction. The algorithm adopts a breadth-first *apriori*-like strategy (Agrawal and Srikant, 1994), similar in spirit to the one developed in Apostolico et al. (2009), using maximal solid blocks as building blocks. Indeed, an important property of maximal dense patterns, which we will exploit in our mining strategy, is that all of their solid blocks are maximal solid blocks. The following proposition extends a similar result holding for arbitrary maximal patterns (Ukkonen, 2007; Pisanti, 2002).

**Proposition 1.** *Let $x$ be a maximal dense pattern with respect to a density threshold $\rho$, and let $b = x[i \ldots j]$ be any solid block in $x$ such that $x[i-1] = x[j+1] = \circ$, for $j \geq i$. Then, $b$ is a maximal solid block.*

**Proof.** Let us assume by contradiction that $b$ is not a maximal solid block. This means that there must exist another solid block $b' \neq b$ which subsumes $b$, that is, such that $f(b) = f(b')$ and there exists $\ell$ with $0 \leq \ell \leq |b'| - |b|$, for which $b = b'[\ell \ldots \ell + |b| - 1]$. Note that it must be $|b'| > |b|$, hence we have that $\ell > 0$ or $\ell + |b| < |b'|$. W.l.o.g, assume that $\ell > 0$, whence $b'[\ell - 1] \neq \circ$ (the case $\ell + |b| < |b'|$ is analogous). Consider now $x'$, which is equal to $x$ except that the $\circ$ symbol preceding $b$ inside $x$ is replaced by $b'[\ell - 1]$ in $x'$. Note that $x'$ is a dense pattern since $x$ is so. Since $f(x) = f(x')$, because $f(b) = f(b')$, and $x'$ contains $x$ by construction, then $x'$ subsumes $x$, which contradicts the maximality of $x$. ■

MADMX begins by extracting the maximal dense motifs from the maximal extensions of the maximal solid blocks. It then operates by repeatedly combining together, in a suitable fashion, pairs of maximal dense motifs, and extracting less frequent maximal dense motifs from these combinations. The combining operation is called *fusion* and is a key notion for the algorithm. Below, we first define fusion for characters, and then extend it to patterns.

**Definition 4.** *Given three characters $c, c_1, c_2 \in \Sigma \cup \{\circ\}$, we say that $c$ is the fusion of $c_1$ and $c_2$, and write $c = c_1 \bigtriangledown c_2$, if one of the following holds:*

1. $c = c_1 = c_2$;
2. $c_1 = \circ$, $c = c_2 \neq \circ$;
3. $c = c_1 \neq \circ$, $c_2 = \circ$.

Observe that $c_1 \bigtriangledown c_2$ is not defined when $c_1$ and $c_2$ are different solid characters. The above notion of fusion generalizes to patterns as follows.

**Definition 5.** *Given three patterns $x$, $y$, $z$ and an integer $d$, we say that $z$ is the $d$-*fusion of $x$ and $y$, *and write $z = x \bigtriangledown_d y$, if $z$ can be obtained by removing the leading and trailing don't care characters from the pattern $m$ defined as $m[i] = x[i + d] \bigtriangledown y[i]$, for all indices $i$.*

Observe that $x \bigtriangledown_d y$ is defined only when the involved individual character fusions are defined.

The breadth-first strategy adopted by MADMX crucially relies on the following theorem, which highlights the structure of dense motifs.

**Theorem 1.** *Let $x$ be a maximal dense motif with $dc(x) > 0$. Then:*

*(a) there exists a maximal solid block $b \preceq x$ such that $\mathcal{M}(x) = \mathcal{M}(b)$, or*
*(b) there exist two maximal dense motifs $y_1$,$y_2$ such that the following conditions hold:*

- *$\mathcal{M}(x) = \mathcal{M}(y_1 \bigtriangledown_d y_2)$ for some $d$;*
- *there are two maximal solid blocks $b_1$, $b_2$ in $x$ and an integer $\hat{d} > 0$ such that $b_1$ is a maximal solid block in $y_1$, $b_2$ is a maximal solid block in $y_2$, and $b_1 \circ^{\hat{d}} b_2 \preceq y_1 \bigtriangledown_d y_2$*
- *$f(x) < \min\{f(y_1), f(y_2)\}$;*

For the proof of Theorem 1, we need to define another type of pattern combination, namely the operation of *merge* between two patterns, which is similar to the one introduced in Pisanti et al. (2005). Given two

characters $c_1$, $c_2$, we define the operator $\oplus$ between them such that $c_1 \oplus c_2 = \circ$, if $c_1 \neq c_2$, and $c_1 \oplus c_2 = c_1 = c_2$, otherwise.

**Definition 6.** *Given two patterns x, y and an integer d, the d-merge of x and y is the pattern $z = x \oplus_d y$ which can be obtained by removing all leading and trailing don't cares from the pattern m defined as $m[i] = x[i+d] \oplus y[i]$ for all i.*

We want to stress the difference between the notions of merging and $d$-fusing: the merge of two patterns $x$, $y$ is always well defined and more general than $x$, $y$, while the $d$-fusion of $x$, $y$ may not exist and, if it does, is more specific than $x$, $y$.

For the proof of Theorem 1, we also need the property established by the following lemma.

**Lemma 1.** *Let x and y be maximal patterns, and d be an integer such that $z = x \oplus_d y$ is a nonempty pattern. Then z is maximal. Moreover, if $z \neq x$ (resp., $z \neq y$) then $f(z) > f(x)$ (resp., $f(z) > f(y)$).*

**Proof.** First we prove that $z$ is maximal. By contradiction, suppose that this is not the case. Then, there exists a position $i$ such that $z[i] = \circ$ can be replaced by a solid character $c$ without decreasing the frequency of the pattern. (Note that the position of the substitution can be to the left of the first character in $z$ or to the right of the last character in $z$.) Since $z$ is contained both in $x$ and $y$, every occurrence of $x$ and $y$ in the string gives raise to an occurrence of $z$. Hence, every occurrence of $x$ (resp., $y$) in the string, contains $c$ in its $(i+d)$th (resp., $i$th) position. Therefore, by maximality of $x$ and $y$, it must be $z[i] = x[i+d] = y[i] = c$, which is a contradiction. The relations between the frequencies of $x$,$y$ and $z$ follow trivially from their maximality. ■

We are now ready to prove Theorem 1.

**Proof.** Given a pattern $x$ and two nonnegative integers $i \leq j$, let $x^*[i \ldots j]$ denote the pattern obtained by removing all the leading and trailing don't care characters from $x[i \ldots j]$. We first show that there exists an index $s_1$, with $0 < s_1 < |x| - 1$, such that $x[s_1] = \circ$ and both $x^*[0 \ldots s_1 - 1]$ and $x^*[s_1 + 1 \ldots |x| - 1]$ are dense. As a first case, assume that $\delta(x[0 \ldots j]) \geq \rho$, for every $0 \leq j < |x|$, and let $s_1 < |x| - 1$ be the position in $x$ of the rightmost don't care. Thus, $\delta(x[0 \ldots s_1 - 1]) \geq \rho$ and $\delta(x[s_1 + 1 \ldots |x| - 1]) = 1 \geq \rho$. Otherwise, let $s_1 = \min\{s : \delta(x[0 \ldots s]) < \rho\}$. Then, it must be that $x[s_1] = \circ$ and $\delta(x[0 \ldots s_1 - 1]) \geq \rho$. Moreover, since $x$ is dense and $\delta(x[0 \ldots s_1]) < \rho$, we must have $\delta(x[s_1 + 1 \ldots |x| - 1]) \geq \rho$, or otherwise

$$\begin{aligned}\delta(x) &= 1 - \frac{dc(x[0 \ldots s_1]) + dc(x[s_1 + 1 \ldots |x| - 1])}{|x|} \\ &= \frac{(|x[0 \ldots s_1]| - dc(x[0 \ldots s_1])) + (|x[s_1 + 1 \ldots |x| - 1]| - dc(x[s_1 + 1 \ldots |x| - 1]))}{|x|} \\ &< \frac{\rho(|x[0 \ldots s_1]| + |x[s_1 + 1 \ldots |x| - 1]|)}{|x|} = \rho\end{aligned}$$

which contradicts the assumption on $\delta(x)$.

We call the pair of dense patterns $x^*[0 \ldots s_1 - 1]$ and $x^*[s_1 + 1 \ldots |x| - 1]$ a *level-1 decomposition* of $x$ (observe that many such decompositions may exist). Now, consider the following iterative process to obtain a sequence of decompositions of $x$ of increasing level, starting from the level-1 decomposition. Initially, set $i = 1$, $\ell_1 = 0$ and $r_1 = |x| - 1$:

1. If both $x^*[\ell_i \ldots s_i - 1]$ and $x^*[s_i + 1 \ldots r_i]$ have frequency strictly greater than $f(x)$, *or* at least one of $x^*[\ell_i \ldots s_i - 1]$ and $x^*[s_i + 1 \ldots r_i]$ is a solid block with frequency equal to $f(x)$, then the process terminates.
2. Otherwise, let $y = x^*[\ell_{i+1} \ldots r_{i+1}]$ be one (arbitrarily chosen) of $x^*[\ell_i \ldots s_i - 1]$ or $x^*[s_i + 1 \ldots r_i]$ which is not a solid block and has frequency equal to $f(x)$. Since $y$ is dense, there exists an index $s_{i+1}$ such that $\ell_{i+1} < s_{i+1} < r_{i+1}$ and both $x^*[\ell_{i+1} \ldots s_{i+1} - 1]$ and $x^*[s_{i+1} + 1 \ldots r_{i+1}]$ are dense. Call these two patterns the level-$(i+1)$ decomposition of $x$. Set $i = i + 1$ and go to Step 1.

Suppose that the decomposition process ends by finding a solid block $b$ in $x$, hence a maximal solid block by Proposition 1, with $f(b) = f(x)$. Then, $\mathcal{M}(b) = \mathcal{M}(x)$ and the theorem follows. Otherwise, let $j$ be the last

level of the decomposition. Then, $f(x) < \min\{f(x^*[\ell_j \dots s_j - 1]), f(x^*[s_j + 1 \dots r_j])\}$. In the latter case, as explained in Section 2 (after Definition 2), we can determine two maximal dense patterns $y_1$, $y_2$ such that $y_1$ subsumes $x^*[\ell_j \dots s_j - 1]$ and $y_2$ subsumes $x^*[s_j + 1 \dots r_j]$. Therefore, $\mathcal{M}(y_1) = \mathcal{M}(x^*[\ell_j \dots s_j - 1])$, $\mathcal{M}(y_2) = \mathcal{M}(x^*[s_j + 1 \dots r_j])$, and $f(x) < \min\{f(y_1), f(y_2)\}$. Let $b_1$ (resp., $b_2$) be the last (resp., the first) solid block of $x^*[\ell_j \dots s_j - 1]$ (resp., $x^*[s_j + 1 \dots r_j]$). Observe that by construction $b_1$ and $b_2$ are maximal solid blocks and there exists an integer $\hat{d}$ such that $b_1 \circ^{\hat{d}} b_2$ is a substring of $x$.

Next, we show that there exists an integer $d$ such that the $d$-fusion $y_1 \bigtriangledown_d y_2$ is well defined, contains $b_1 \circ^{\hat{d}} b_2$, and $\mathcal{M}(y_1 \bigtriangledown_d y_2) = \mathcal{M}(x)$. We proceed as follows. First, we show that $\mathcal{M}(x)$ contains both $y_1$ and $y_2$. To this end, let us "align" $\mathcal{M}(x)$ and $\mathcal{M}(y_1)$ with respect to the occurrence of $x^*[\ell_j \dots s_j - 1]$, which is contained in both, and let $p$ the integer such that $\mathcal{M}(x)[i + p]$ is aligned with $\mathcal{M}(y_1)[i]$. Now, assume for the sake of contradiction, that there exists an index $j$ such that $\mathcal{M}(y_1)[j]$ corresponds to a position of $y_1$, it is solid and $\mathcal{M}(y_1)[j] \neq \mathcal{M}(x)[j + p]$. This implies that $z = \mathcal{M}(x) \oplus_p \mathcal{M}(y_1) \neq \mathcal{M}(y_1)$. Moreover, $z$ contains $x^*[\ell_j \dots s_j - 1]$, and, by Lemma 1, it is maximal and has frequency strictly greater than $f(y_1)$, which is impossible because we have chosen $y_1$ such that $\mathcal{M}(x^*[\ell_j \dots s_j - 1]) = \mathcal{M}(y_1)$ and therefore $f(x^*[\ell_j \dots s_j - 1]) = f(y_1)$. A similar argument shows that $\mathcal{M}(x)$ contains $y_2$.

Since $y_1$ and $y_2$ are contained in $\mathcal{M}(x)$, there must exist a $d$ such that $y_1 \bigtriangledown_d y_2$ is also contained in $\mathcal{M}(x)$, and can be aligned with $\mathcal{M}(x)$ in such a way to match the blocks $b_1$ and $b_2$ of $y_1$ and $y_2$ with the corresponding blocks in $\mathcal{M}(x)$. Moreover, $f(y_1 \bigtriangledown_d y_2) \geq f(\mathcal{M}(x)) = f(x)$. However, since $y_1 \bigtriangledown_d y_2$ contains both $x^*[\ell_j \dots s_j - 1]$ and $x^*[s_j + 1 \dots r_j]$, it contains also $x^*[\ell_j \dots r_j]$, which, by the decomposition process, has frequency equal to $f(x)$. Therefore, $f(y_1 \bigtriangledown_d y_2) \leq f(x)$, and the theorem follows since $f(y_1 \bigtriangledown_d y_2) = f(x)$. ■

In essence, Theorem 1 guarantees that we can find any maximal dense motif $x$ either within $\mathcal{M}(b)$, for some maximal solid block $b$, or by $d$-fusing two higher-frequency maximal dense motifs $y_1$, $y_2$, for some $d$, finding $z = \mathcal{M}(y_1 \bigtriangledown_d y_2)$ and then possibly "trimming" $z$ on both sides to obtain $x$. Also, the theorem shows that in the latter case the trimmed sequence must contain at least one maximal solid block $b_1$ of $y_1$ and one maximal solid block $b_2$ of $y_2$. Moreover, we can disregard those $d$-fusions $y_1 \bigtriangledown_d y_2$ for which no pair of maximal solid blocks $b_1$ of $y_1$ and $b_2$ of $y_2$ exists such that $b_1 \circ^{\hat{d}} b_2$ is contained in $y_1 \bigtriangledown_d y_2$ for some $\hat{d} > 0$.

Algorithm MADMX, whose pseudocode is reported in Figure 1, implements the strategy inspired by Theorem 1. It employs three (initially empty) sets *previous*, *current*, and *next*. In line 2, the algorithm first stores the maximal solid blocks $b$ in $s$ for the given frequency $\sigma$ in the set *blocks* (see Section 2). Then, it extracts all of the appropriate maximal dense motifs from $\mathcal{M}(b)$ in lines 3–6, using the function extractMaximalDense, as implied by Theorem 1(a). Finally, lines 7–16 implement the strategy as implied by Theorem 1(b). (In line 10, a fusion $y_1 \bigtriangledown_d y_2$ is considered *valid* if it satisfies the second property of Theorem 1(b).) In practice, the function extractMaximalDense($\mathcal{M}(x)$) produces all the maximal dense patterns contained in $\mathcal{M}(x)$ and that contains $x$. The second property of Theorem 1(b) guarantees that even with this restriction all the maximal dense motifs will be produced in output.

### 3.1. Efficient implementation of MADMX

An important issue for the efficiency of MADMX is that it needs to compute the exact frequency of each generated pattern. For what concerns the fusion operation, we observe that $x_1 \bigtriangledown_d x_2$ is valid if and only if there exist $\ell_1 \in \mathcal{L}_{x_1}$ and $\ell_2 \in \mathcal{L}_{x_2}$ such that $\ell_1 - \ell_2 = d$; thus, a simple computation on the pairs $(\ell_1, \ell_2) \in \mathcal{L}_{x_1} \times \mathcal{L}_{x_2}$ is sufficient to yield the frequencies of all the valid fusions of two patterns.

Let $z = x_1 \bigtriangledown_d x_2$. Observe that while the exact frequency of a maximal dense pattern $w$ extracted from $\mathcal{M}(z)$ is equal to $f(z)$ in case $w$ contains $z$ in its entirety, for a maximal dense pattern $w$ extracted from $\mathcal{M}(z)$ which does not contain $z$ we can only conclude that $f(w) \geq f(z)$. In this latter case, naively determining the actual frequency may be computationally expensive. Therefore, in the course of the algorithm we generate two classes of maximal dense motifs: those whose exact frequencies are known, and those for which only a lower bound to their frequencies is known.

We modify function extractMaximalDense so to label each generated motif as either *final* or *tentative* depending on whether its frequency is exact or only estimated through a lower bound. Note that for each tentative dense motif $w$ Theorem 1 ensures that there exist two maximal dense motifs $x_1$, $x_2$ and a valid $d$-fusion $x_1 \bigtriangledown_d x_2$ such that $f(\mathcal{M}(x_1 \bigtriangledown_d x_2)) = f(w)$. Hence, we are assured that if $w$ is generated from $x_1$ and

**Algorithm** MADMX()

**Input**: String $s$, frequency threshold $\sigma$, density threshold $\rho$

**Output**: Maximal dense motifs

```
1  previous ← ∅, current ← ∅, next ← ∅ ;
2  blocks ← {maximal solid blocks of s with frequency ≥ σ};
3  for each b ∈ blocks do
4    find M(b) ;
5    DM ← extractMaximalDense(M(b));
6    for each x ∈ DM do current ← current ∪ {x};
7  while current ≠ ∅ do
8    for each x1 ∈ current do
9      for each x2 ∈ previous ∪ current do
10       for each d s.t. z = x1 ▽d x2 is a valid d-fusion do
11         find M(z);
12         DM ← extractMaximalDense(M(z));
13         for each x ∈ DM do
14           if f(x) ≥ σ and x ∉ previous ∪ current then next ← next ∪ {x};
15   previous ← previous ∪ current;
16   current ← next; next ← ∅;
17 return previous;
```

**FIG. 1.** Pseudocode of algorithm MADMX.

$x_2$ and the true frequencies of $x_1$ and $x_2$ are known, the estimated frequency for $w$ is also the true one, even if $w$ is labeled as tentative.

Note that algorithm MADMX may generate the same maximal dense motif $w$ several times, from fusions of different pairs of patterns. The algorithm can be modified in such a way that each time a tentative motif $w$ is (re)generated, if its exact frequency $f(w)$ is inferred then $w$ is (re)labeled final, otherwise, the lower bound to $f(w)$ is updated, if necessary. A further modification to the algorithm consists in requiring that $x_1$ and $x_2$ in Lines 8 and 9 of the pseudocode be final. Whenever the set *current* contains no final motifs, we can label as final the motif in *current* with the highest lower bound to its frequency, and continue with the generation. This is justified by the following proposition.

**Proposition 2.** *Suppose that at some point during the execution of* MADMX *all motifs in the set current are labeled tentative, and let w be the motif belonging to current with the highest lower bound $\ell$ on its frequency. Then $f(w) = \ell$.*

**Proof.** For the sake of contradiction, assume that $f(w) \neq \ell$. In particular, it must be $f(w) > \ell$. From Theorem 1, we know that there must be two dense motifs $x_1$, $y_1$ with $\min\{f(x_1), f(y_1)\} > f(w)$ and an integer $d$ such that $w$ can be obtained, with its exact frequency, from $\mathcal{M}(x_1 \bigtriangledown_d y_1)$. If both $x_1$ and $y_1$ have already been moved to the *previous* list from Algorithm MADMX, we have $f(w) = \ell$. The only possibility is then that at least one of $x_1$ and $y_1$ has not been moved to *previous*. Let $x_1$ be this dense motif. Then $x_1$ is either a *tentative* motif or has not been generated by any fusion yet. Applying the same reasoning to $x_1$, we have that there exists two dense motifs $x_2$, $y_2$ such that at least one of them (let say $x_2$) has not been put in *previous*, $\min\{f(x_2), f(y_2)\} > f(x_1)$ and $x_1$ can be obtained, with its real frequency, from a valid fusion of $x_2$, $y_2$. Iterating this reasoning, we can find a sequence $x_1, x_2, \ldots$ of dense motifs such that

1. $\forall i$, $x_i$ has not been put in *previous*,
2. $f(x_{i+1}) > f(x_i)$, and
3. $x_i$ is derived from the fusion of $x_{i+1}$ with another pattern.

Theorem 1 implies that this sequence must be finite, and that the last element of this sequence, $\tilde{x}$, is either a solid block or can be found in the maximal extension of a solid block. Therefore, $\tilde{x}$ has been generated by the algorithm (lines 3–5) with its correct frequency, thus it is in *previous*, that is a contradiction. ■

A crude upper bound on the running time of MADMX can be derived by observing that, for each pair of dense maximal motifs in output, the time spent during all the operations concerning that pair is (naively) $O(n^3)$, where $n$ is the length of the input string. If $P$ patterns are produced in output, the overall time complexity is $O(n^3P^2)$.

## 4. EXPERIMENTAL VALIDATION OF MADMX

We developed a prototype-based implementation of MADMX in C++ also including an additional feature which eliminates, from the set of initial maximal solid blocks, those shorter than a given threshold $min_\ell$. The purpose of the latter heuristics is to speed up motif generation driving it towards the discovery of (possibly) more significant motifs, with the exclusion of spurious, low-complexity ones. (The code is available for download at `www.dei.unipd.it/wdyn/?IDsezione=4534.`)

We performed two classes of experiments to evaluate how significant is the set of motifs found using our approach. The first class of experiments is described in Section 4.1. It compares the motifs extracted by MADMX with the known biological repetitions that are available in `RepBase` (Jurka et al., 2005)—a very popular genomic database—using the REPEATMASKER tool described in Smit et al. (1996). The second class of experiments is described in Section 4.2 and aims at comparing the motifs extracted by MADMX with those extracted by VARUN using the same $z$-score metric employed in Apostolico et al. (2009) for assessing their relative statistical significance.

### 4.1. Evaluating significance by known biological repetitions

`RepBase` (Jurka et al., 2005) is one of the largest repositories of prototypic sequences representing repetitive DNA from different eukaryotic species, collected in several different ways. `RepBase` is used as a reference collection for masking and annotating repetitive DNA through popular tools such as REPEATMASKER (Smit et al., 1996). The latter screens an input DNA sequence $s$ for simple repeats and low complexity portions, and interspersed repeats using `RepBase`. Sequence comparisons are performed through Smith-Waterman scoring (Smith and Waterman, 1981). REPEATMASKER returns a detailed annotation of the repeats occurring in $s$, and a modified version of $s$ in which all of the annotated repeats are masked by a special symbol (N or X). With the current version of `RepBase`, on average, almost 50% of a human genomic DNA sequence will be masked by the program (Smit et al., 1996).

Most of the interspersed repeats found by REPEATMASKER belong to the families called SINE/ALU and LINE/L1: the former are *Short INterspersed Elements* that are repetitive in the DNA of eukaryotic genomes (the Alu family in the human genome); the latter are *Long Interspersed Nucleotide Elements*, which are typically highly repeated sequences of 6K–8K bps, containing RNA polymerase II promoters. The LINE/L1 family forms about 15% of the human genome.

We have conducted an experimental study using MADMX and REPEATMASKER on *Human Glutamate Metabotropic Receptors* HGMR 1 (410277 bps) and HGMR 5 (91243 bps) as input sequences. We have downloaded the sequences from the March 2006 release of the UCSC Genome database (`http://genome.ucsc.edu`), with genomic coordinates hg18_dna range=chr6:146385472–146805427 (HGMR 1) and hg18_dna range=chr11:87872389–88443761 (HGMR 5). REPEATMASKER version was open-3.2.7, sensitive mode, with the query species assumed to be homologous; it ran using blastp version 2.0a19MP-WashU, and `RepBase` update 20090120.

The experiments to assess the biological significance of the maximal dense motifs extracted by MADMX involved three separate stages.

In the first stage, we ran REPEATMASKER on the input sequences HGMR 1 and HGMR 5, searching for interspersed repeats using `RepBase`. We considered the summary (tbl file) provided by REPEATMASKER and one of its output files (.out) containing the list of found repeats: for each occurrence, the .out file provides the substring $s[i \ldots j]$ of the input sequence $s$ which is locally aligned with (a substring of) the repeat and is annotated with extra information (which part of the repeat is aligned, its Smith-Waterman score, and so on).

In the second stage, we ran MADMX on the same DNA sequences, with density threshold $\rho = 0.8$, frequency threshold $\sigma = 4$, and $\min_\ell = 15$. In order to filter out simple repeats and low complexity portions, which are dealt with by REPEATMASKER without resorting to RepBase, we modified MADMX eliminating periodic maximal solid blocks (with short periods), which are the seeds of simple repeats. Then, we identified the occurrences of the motifs returned by MADMX in the input sequences, using REPEATMASKER as a pattern matching tool (i.e., replacing RepBase with the set of motifs returned by MADMX as the database of known repeats). The underlying idea behind this use of REPEATMASKER was to employ the same local alignment algorithms, so to make the comparison fairer.

In the third stage, we cross-checked the intervals associated with the occurrences of the RepBase repeats against those associated with the occurrences of our motifs. Specifically, we mapped the motif occurrences in $s$ (seen as intervals $[i' \ldots j']$ found by MADMX) into the repeats (seen as intervals $[i \ldots j]$ found by REPEATMASKER) using an approximate notion of interval inclusion. Specifically, a motif occurrence $[i' \ldots j']$ is mapped into a repeat $[i \ldots j]$ whenever $[i', j'] \subseteq [i - \delta, j + \delta]$ and $[i, j] \subseteq [i' - \delta, j' + \delta]$, for a very small constant $\delta$. Surprisingly, through the above mapping MADMX was able to identify and characterize *all* of the intervals of the known SINE/ALU repeats in HGMR 1 and HGMR 5 (respectively, 56 repeats plus an extra unclassified one for HGMR 1, and 20 repeats plus an extra unclassified one for HGMR 5). The remaining occurrences of the motifs permitted to identify 29 repeats out of 78 of the LINE/L1 family in HGMR 1.

### 4.2. Evaluating significance by statistical z-score ranking

The z-score is the measure of the distance in standard deviations of the outcome of a random variable from its expectation. Consider a DNA sequence $s$ of length $n$ as if it were generated by a stationary i.i.d. source with equiprobable symbols. An approximation to the z-score for a motif of length $m$ that contains $c$ solid characters and appears $f$ times in $s$ is given by $Z = \frac{f - (n-m+1)\times p}{\sqrt{(n-m+1)\times p \times (1-p)}}$, where $p = (1/4)^c$. This metric was used in Apostolico et al. (2009) to assess the significance of the motifs extracted by VARUN and to rank them in the output.

We employed the code for VARUN provided by the authors to extract the rigid motifs from the DNA sequences analyzed in Apostolico et al. (2009). We then ran MADMX on the same sequences (provided to us by the authors of Apostolico et al., 2009) using the same frequency parameters, and setting the minimum density threshold $\rho$ in such a way to obtain a comparable yet smaller output size. In this fashion, we tested the ability of MADMX to produce a succinct yet significant set of motifs, by virtue of its more flexible notion of density.

The results are shown in Table 1. For VARUN we used $D = 1$, thus allowing at most one don't care between two solid characters, and ran MADMX with $min_\ell = 1$, so to obtain the *complete* family of maximal dense motifs. In the table, there is a row for each sequence (identified in the first column). Each sequence, whose total length is reported in the second column, is obtained as the concatenation of a number of smaller subsequences, reported in the third column. On each sequence, both tools were run with the same frequency threshold $\sigma$, and the table reports for both the output size in terms of the number of motifs returned and the execution time in seconds. Also, for MADMX, the table reports the density threshold $\rho$ used in each experiment.

For each experiment, we compared the best top-$m$ z-scores, with $m = 10$, 50, and 100, as follows. Note that, in general, the top-$m$ motifs found by MADMX and VARUN differ. Thus, we let $z^i_M$ (resp., $z^i_V$) be the

Table 1. Results of the Comparison with VARUN

| | | | VARUN | | | MADMX | | | Best top-m z-scores | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| *Name* | *Length* | *No.* | $\sigma$ | $\lvert Output \rvert$ | *Time* | $\rho$ | $\lvert Output \rvert$ | *Time* | $m = 10$ | $m = 50$ | $m = 100$ | $m^*$ | $\hat{m}$ |
| ace2 | 500 | 1 | 2 | 1866 | 3s | 0.7 | 1762 | 18s | 10 | 50 | 100 | 1571 | 1067 |
| ap1 | 500 | 1 | 2 | 1555 | 1s | 0.7 | 1304 | 5s | 10 | 50 | 100 | 392 | 13 |
| gal4 | 3000 | 6 | 4 | 9764 | 12s | 0.67 | 7606 | 67s | 10 | 49 | 99 | 16 | 16 |
| gal4(*) | 3000 | 6 | 4 | 9764 | 12s | 0.65 | 11733 | 191s | 10 | 50 | 100 | 9764 | 301 |
| uasgaba | 1000 | 2 | 2 | 4586 | 30s | 0.70 | 4194 | 90s | 10 | 50 | 100 | 175 | 175 |

z-score of the $i$th motif in decreasing z-score order obtained by MADMX (resp., VARUN). For each $m$, the table reports how many times it was $z_M^i > z_V^i$, for $1 \leq i \leq m$. Also, column $m^*$ (resp., column $\hat{m}$) gives the maximum $m$ such that $z_M^i \geq z_V^i$ (resp., $z_M^i > z_V^i$) for every $1 \leq i \leq m$.

Even when MADMX is calibrated to yield a slightly smaller output, the quality of the motifs extracted, as measured by the z-score, is higher than those output by VARUN. Indeed, for sequences `ace2` and `uasgaba`, a very large prefix of the top-ranked motifs extracted by MADMX features strictly greater z-scores of the corresponding top-ranked ones extracted by VARUN. For all of the four sequences, at least the thirteen top-ranked motifs have this property. To shed light on the slightly worse performance of MADMX on `gal4`, we re-ran MADMX with a different density threshold, so to obtain a slightly larger output (see row gal4$^{(*)}$). In this case, the top-301 motifs extracted by MADMX have z-score strictly greater than the corresponding motifs extracted by VARUN, while the execution time still remains acceptable.

For all runs, the top z-score of a motif discovered by MADMX is considerably higher than the one returned by VARUN. Specifically, on `ace2` our best z-score is 387,763 versus 12,027 of VARUN; on `ap1`, we have 12,027 versus 1,490; on `gal4` it is 75 versus 28; on `gal4(*)` it is 150 versus 28; on `uasgaba` we have 134,532 versus 67,059. This reflects the high selectivity of MADMX, which is to be attributed mostly to adoption of a more flexible density constraint.

We must remark that MADMX (in its current nonoptimized version) is slower than VARUN, but it still runs in time acceptable from the point of view of a user. To further investigate the tradeoff between execution time and significance of the discovered motifs, we repeated the experiments running MADMX with $\min_\ell = 2$ and $\rho = 0.65$, for all sequences. The running time of MADMX was almost halved, while the small output produced still featured high quality. In fact, for sequences `ace2`, `ap1`, and `uasgaba` the top-100 motifs extracted by MADMX have z-score greater or equal than the corresponding ones returned by VARUN.

We also have attempted a comparison between VARUN and MADMX on longer sequences (such as HGMR 1) at higher frequencies (since, unfortunately, VARUN does not seem to be able to handle low frequencies on very long sequences). Even allowing a higher number of don't cares between solid characters ($D = 2$) for the motifs of VARUN, all of the top-$m$ z-scores featured by the motifs extracted by MADMX are greater than or equal to the corresponding scores in the ranking of VARUN, with $m$ reaching the size of VARUN's output. In fairness, we remark that VARUN was designed to work at its best on protein sequences, while MADMX's main target are DNA sequences. Hence, these two tools should be regarded as complementary. Moreover, VARUN has the advantage of retrieving flexible motifs, while MADMX focuses only on rigid ones.

## 5. CONCLUSION

In this article, we introduced the notion of density to single out the most relevant motifs during pattern discovery in biological sequences. We showed how to perform the efficient extraction of maximal dense motifs and experimented the corresponding software tool MADMX on real data sets. The efficiency of our approach relies on the newly defined operation of fusion, which avoids the generation of nonmaximal motifs during the intermediate steps of the extraction. The experimental results give evidence of the efficiency and the quality of the motifs returned by MADMX.

## ACKNOWLEDGMENTS

We wish to thank Alberto Apostolico and Matteo Comin for providing the code and the sequences used in Section 4.2, and giving valuable insights on VARUN; Ben Raphael for suggesting the use of REPEATMASKER; and Roberta Mazzucco and Francesco Peruch for coding MADMX. A preliminary version of this work has been presented in WABI 2010. Support for R.G., A.P., N.P., and G.P. was provided, in part, by MIUR of Italy under Project AlgoDEEP prot. 2008TFBWL4. Support for A.P. and G.P. was provided, in part, by the University of Padova under the Strategic Project STPD08JA32 and Project CPDA099949/09. Support for E.U. was provided, in part, by NSF awards IIS-0325838 and DMI-0600384, and ONR Award N000140610607. Part of this work was done while F.V. was affiliated to Dipartimento di Ingegneria dell'Informazione, Università di Padova, Italy.

## DISCLOSURE STATEMENT

No competing financial interests exist.

## REFERENCES

Agrawal, R., and Srikant, R. 1994. Fast algorithms for mining association rules. *Proc. 20th VLDB* 487–499.

Apostolico, A., Comin, M., and Parida, L. 2009. VARUN: discovering extensible motifs under saturation constraints. *IEEE Trans. Comput. Biol. Bioinform.* http://doi.ieeecomputersociety.org/10.1109/TCBB.2008.123.

Apostolico, A., and Parida, L. 2004. Incremental paradigms of motif discovery. *J. Comput. Biol.* 11, 15–25.

Apostolico, A., and Tagliacollo, C. 2007. Optimal offline extraction of irredundant motif bases. *Lect. Notes Comput. Sci.* 4598, 360–371.

Apostolico, A., and Tagliacollo, C. 2008. Incremental discovery of the irredundant motif bases for all suffixes of a string in $O(n^2 \log n)$ time. *Theor. Comput. Sci.* 408, 106–115.

Arimura, H., and Uno, T. 2008. Mining maximal flexible patterns in a sequence. *Lect. Notes Comput. Sci.* 4914, 307–317.

Jurka, J., Kapitonov, V.V., Pavlicek, A., et al. 2005. Repbase update: a database of eukaryotic repetitive elements. *Cytogenet. Genome Res.* 110, 462–467.

Morris, M., Nicolas, F., and Ukkonen, E. 2008. On the complexity of finding gapped motifs. *CoRR* abs/0802.0314.

Parida, L. 2000. Some results on flexible-pattern discovery. *Lect. Notes Comput. Sci.* 1848, 33–45.

Parida, L. 2008. *Pattern Discovery in Bioinformatics*. Chapman & Hall/CRC, Boca Raton, FL.

Pisanti, N. 2002. Segment-based distances and similarities in genomic sequences [Ph.D. dissertation]. University of Pisa, Italy.

Pisanti, N., Crochemore, M., Grossi, R., et al. 2005. Bases of motifs for generating repeated patterns with wild cards. *IEEE Trans. Comput. Biol. Bioinform.* 2, 40–50.

Rigoutsos, I., and Floratos, A. 1998. Combinatorial pattern discovery in biological sequences: the TEIRESIAS algorithm. *Bioinformatics* 14, 55–67.

Saha, S., Bridges, S., Magbanua, Z.V., et al. 2008. Empirical comparison of *ab initio* repeat finding programs. *Nucleic Acids Res.* 36, 2284–2294.

Smit, A.F.A., Hubley, R., and Green, P. 1996. *RepeatMasker Open-3.0*. Available at: www.repeatmasker.org. Accessed January 15, 2011.

Smith, T.F., and Waterman, M.S. 1981. Identification of common molecular subsequences. *J. Mol. Biol.* 147, 195–197.

Ukkonen, E. 2007. Structural analysis of gapped motifs of a string. *Lect. Notes Comput. Sci.* 4708, 681–690.

Address correspondence to:
*Dr. Fabio Vandin*
*Department of Computer Science*
*Brown University*
*Providence, RI 02906*

*E-mail:* vandinfa@cs.brown.edu