A Basis of Tiling Motifs for Generating Repeated Patterns and its Complexity for Higher Quorum*

N. Pisanti[1], M. Crochemore[2,3] **, R. Grossi[1], and M.-F. Sagot[4,3] ***

[1] Dipartimento di Informatica, Università di Pisa, Italy
{pisanti,grossi}@di.unipi.it
[2] Institut Gaspard-Monge, University of Marne-la-Vallée, France
Maxime.Crochemore@univ-mlv.fr
[3] INRIA Rhône Alpes, France Marie-France.Sagot@inria.fr
[4] King's College London, UK

Abstract. We investigate the problem of determining the basis of motifs (a form of repeated patterns with don't cares) in an input string. We give new upper and lower bounds on the problem, introducing a new notion of basis that is provably smaller than (and contained in) previously defined ones. Our basis can be computed in less time and space, and is still able to generate the same set of motifs. We also prove that the number of motifs in all these bases grows exponentially with the quorum, the minimal number of times a motif must appear. We show that a polynomial-time algorithm exists only for fixed quorum.

1 Introduction

Identifying repeated patterns in strings is a computationally-demanding task on the large data sets available in computational biology, data mining, textual document processing, system security, and other areas; for instance, see [6]. We consider patterns with *don't cares* in a given string s of n symbols drawn over an alphabet Σ. The don't care is a special symbol '∘' matching any symbol of Σ; for example, pattern `T∘E` matches both `TTE` and `TEE` inside $s =$ `COMMITTEE` (note that a pattern cannot have a don't care at the beginning or at the end, as this is not considered informative). Contrarily to string matching with don't cares, the pattern `T∘E` is not given in advance for searching s. Instead, the patterns with don't cares appearing in s are *unknown* and, as such, have to be *discovered* and *extracted* by processing s efficiently. In our example, `T∘E` and `M∘∘T∘E` are among the patterns appearing repeated in `COMMITTEE`. In this paper we focus on finding the patterns called *motifs*, which appear at least q times in s for an

* The full version of this paper is available in [11] as technical report TR-03-02.

** Supported by CNRS action AlBio, NATO Sc. Prog. PST.CLG.977017, and Wellcome Trust Foundation.

*** Supported by CNRS-INRIA-INRA-INSERM action BioInformatique and Wellcome Trust Foundation.

input parameter $q \geq 2$ called the *quorum*. Different formulations in the known literature address the problem of detecting motifs in several contexts, revealing its algorithmic relevance. Unfortunately, the complexity of the algorithms for motif discovery may easily become exponential due to the explosive growth of the motifs in strings, such as in the artificial string $\mathtt{A} \cdots \mathtt{ATA} \cdots \mathtt{A}$ (same number of As on both sides of T) generating many motifs with As intermixed with don't cares, and in other "real" strings over a small alphabet occurring in practice, e.g., DNA sequences. Some heuristics try to alleviate this drawback by reducing the number of interesting motifs to make feasible any further processing of them, but they cannot guarantee sub-exponential bounds in the worst case [7].

In this paper, we explore the algorithmic ideas behind motif discovery while getting some insight into their combinatorial complexity and their connections with string algorithmics. Given a motif x for a string s of length n, we denote the set of positions on s at which the occurrences of x start by $\mathcal{L}_x \subseteq [0\,.\,.n-1]$, where $|\mathcal{L}_x| \geq q$ holds for the given quorum $q \geq 2$. We single out the *maximal* motifs x, informally characterized as satisfying $|\mathcal{L}_x| \neq |\mathcal{L}_y|$ for any other motif y more *specific* than x, i.e., obtained from x by adding don't cares and alphabet letters or by replacing one or more don't cares with alphabet letters. In other words, x appears in y but x occurs in s more times than y does, which is considered informative for discovering the repetitions in s. For example, $\mathtt{M{\circ}{\circ}T{\circ}E}$ is maximal in COMMITTEE for $q = 2$ while $\mathtt{M{\circ}{\circ}{\circ}{\circ}E}$ and $\mathtt{T{\circ}E}$ are not maximal since $\mathtt{M{\circ}{\circ}T{\circ}E}$ is more specific with the same number of occurrences. Maximality provides an intuitive notion of relevance as each maximal motif x indirectly represents all non-maximal motifs z that are less specific than it. Unfortunately, this property does not bound significantly the number of maximal motifs. For example, $\mathtt{A} \cdots \mathtt{ATA} \cdots \mathtt{A}$ contains an exponential number of them for $q = 2$ (see Section 2). A further requirement on the maximal motifs is the notion of *irredundant* motifs ([7]). A maximal motif x is redundant if there exist maximal motifs $y_1, \ldots, y_k \neq x$ such that the set of occurrences of x satisfies $\mathcal{L}_x = \mathcal{L}_{y_1} \cup \ldots \cup \mathcal{L}_{y_k}$; it is irredundant otherwise. The set of occurrences of a redundant motif can be *covered* by other sets of occurrences while that of an irredundant motif is *not* the union of the sets of occurrences of other maximal motifs. The *basis of the irredundant motifs* of string s with quorum q is the set of irredundant motifs in s. Informally speaking, a basis can generate all the motifs by simple rules and can be expressed mathematically in the algebraic sense of the term. According to Parida et al. [7], what makes interesting the irredundant motifs is that their number is always upper bounded by $3n$ independently of *any chosen* $q \geq 2$; moreover, they can be found in $O(n^3 \log n)$ time by this bound, notwithstanding the possibly exponential number of maximal motifs that are candidates for the basis.

Our results: We study the complexity of finding the basis of motifs with novel algorithms to represent all motifs succinctly. We show that, in the worst case, there is an infinite family of strings for which the basis contains $\Omega(n^2)$ irredundant motifs for $q = 2$ (see Section 2). This contradicts the upper bound of $3n$ for any $q \geq 2$ given in [7] as shown (in the Appendix of [11] we give a counterexample to its charging scheme, which crucially relies on a lemma that is not valid). As a result, the bound of $O(n^3 \log n)$ time in [7] for any q does not hold since it relies on the upper bound of $3n$, thus leaving open the problem of discovering a basis in polynomial time for any q. We also introduce a new definition called *basis of the tiling motifs* of string s with quorum q. The condition for tiling motifs is stronger than that of irredundancy. A maximal motif x is *tiled* if there exist maximal motifs $y_1, \ldots, y_k \neq x$ such that the set of occurrences of x satisfies $\mathcal{L}_x = (\mathcal{L}_{y_1} + d_1) \cup \ldots \cup (\mathcal{L}_{y_k} + d_k)$ for some integers $d_1, \ldots, d_k$; it is tiling otherwise. Note that the motifs $y_1, \ldots, y_k$ are not necessarily distinct and the union of their occurrences is taken after displacing them by $d_1, \ldots, d_k$, respectively. Since a redundant motif is also tiled with $d_1 = \cdots = d_k = 0$, a tiling motif is surely irredundant. Hence the basis for the tiling motifs is included in the basis for irredundant motifs while both of them are able to generate the *same* set of motifs with mechanical rules. Although the definition of tiling motifs is derived from that of irredundant ones, the difference is much more substantial than it may appear. The basis of tiling motifs is symmetric, namely, the tiling motifs of $\widetilde{s}$ (the string s in reversed order) are the reversed tiling motifs of s whereas the irredundant motifs for strings s and $\widetilde{s}$ are apparently unrelated, unlike the entropy and other properties related to the repetitions in strings. Moreover, the number of tiling motifs can be provably upper bounded in the worst case by $n-1$ for $q = 2$ and they occur in s for a total of $2n$ times at most, whereas we demonstrate that there can be $\Omega(n^2)$ irredundant motifs. We give more details in Section 3, and we also discuss in the full paper [11] how to find the longest motifs with a limited number of don't cares. Finally, in Section 4, we reveal an *exponential* dependency on the quorum q for the number of motifs, both for the basis of irredundant motifs and for the basis of tiling motifs, which was unnoticed in previous work. We prove that there is an infinite family of strings for which the basis contains at least $\binom{\frac{n-1}{2}-1}{q-1} = \Omega\left(\frac{1}{2^q}\binom{n-1}{q-1}\right)$ tiling (hence, irredundant) motifs. Hence, *no* worst-case polynomial-time algorithm can exist for finding the basis with *arbitrary* values of $q \geq 2$. Nonetheless, we can prove that the tiling motifs in our basis are less than $\binom{n-1}{q-1}$ in number and occur in s a total of $q\binom{n-1}{q-1}$ times at most. For them there exists a pseudo-polynomial algorithm taking $O\left(q^2\binom{n-1}{q-1}^2\right)$ time, which shows that the tiling motifs can be found in polynomial time if and only if the quorum q satisfies either $q = O(1)$ or $q = n - O(1)$ (the latter is hardly meaningful in practice). Experimenting with small strings exhibits a non-constant growth of the basis for increasing values of q up to $O(\log n)$ but larger values of q are possible in the worst case. More experimental analysis of the implementation can be found in [11]. Proofs of all results can also be found in [11].

Related work: As previously mentioned, the seminal idea of basis was introduced by Parida *et al.* [7]. The unpublished manuscript [3] adopted an identical definition of irredundant motifs in the first part. Very recently, Apostolico [2] observed that the $O(n^3)$-time algorithm proposed in the second part of [3] contains an implicit definition different from that of the first part. Namely, in a redundant motif x, the list $\mathcal{L}_x$ can be "deduced" from the union of the others (see also [1]). Note that no formal specification of this alternative definition is however explicited. Applications of the basis of *repeated* patterns (with just $q = 2$) to data compression are described in [4]. Tiling motifs can be employed in this context because of their linear number of occurrences in total.

The idea of the basis was also explored by Pelfrêne *et al.* [8, 9], who introduced the notion of *primitive motifs*. They gave two alternative definitions claimed to be equivalent, one definition reported in the two-page abstract accompanying the poster and the other in the poster itself. The basis defined in the poster is not symmetric and is a superset of the one presented in this paper. On the other hand, the definition of primitive motifs given in the two-page abstract is somehow equivalent to that given in this paper and introduced independently in our technical report [10]. Because of the lower bounds proved in this paper, the algorithm in [9] is exponential with respect to q.

The problem of finding a polynomial-size basis for higher values of q remains unsolved.

2 Irredundant Motifs: The Basis and its Size for $q = 2$

We consider strings that are finite sequences of letters drawn from an alphabet Σ, whose elements are also called *solid characters*. We introduce an additional letter (denoted by $\circ$ and called *don't care*) that does not belong to Σ and matches any letter. The length of a string t with don't cares, denoted by $|t|$, is the number of letters in t, and $t[i]$ indicates the letter at position i in t for $0 \le i \le |t| - 1$ (hence, $t = t[0]t[1] \cdots t[|t| - 1]$ also noted $t[0 \,.\,.\, |t| - 1]$). A *pattern* is a string in $\Sigma \cup \Sigma(\Sigma \cup \{\circ\})^* \Sigma$, that is, it starts and ends with a solid character. The pattern occurrences are related to the *specificity relation* $\preceq$. For individual characters $\sigma_1, \sigma_2 \in \Sigma \cup \{\circ\}$, we have $\sigma_1 \preceq \sigma_2$ if $\sigma_1 = \circ$ or $\sigma_1 = \sigma_2$. Relation $\preceq$ extends to strings in $(\Sigma \cup \{\circ\})^*$ under the convention that each string t is implicitly surrounded by don't cares, namely, letter $t[j]$ *is* $\circ$ when $j < 0$ or $j \ge |t|$. In this way, v is *more specific* than u (shortly, $u \preceq v$) if $u[j] \preceq v[j]$ for any integer j. We also say that u *occurs at position* ℓ *in* v if $u[j] \preceq v[\ell + j]$, for $0 \le j \le |u| - 1$. Equivalently, we say that u *matches* $v[\ell] \cdots v[\ell + |u| - 1]$. For the input string $s \in \Sigma^*$ with $n = |s|$, we consider the occurrences of arbitrary patterns x in s. The *location list* $\mathcal{L}_x \subseteq [0 \,.\,.\, n - 1]$ denotes the set of all the positions on s at which x occurs. For example, the location list of $x =$ `T∘E` in $s =$ `COMMITTEE` is $\mathcal{L}_x = \{5, 6\}$.

Definition 1 (motif). *Given a parameter $q \ge 2$ called* quorum*, we say that pattern x is a* motif *according to s and q if $|\mathcal{L}_x| \ge q$.*

Given any location list $\mathcal{L}_x$ and any integer d, we adopt the notation $\mathcal{L}_x + d = \{\ell + d \mid \ell \in \mathcal{L}_x\}$ for indicating the occurrences in $\mathcal{L}_x$ "displaced" by the offset d.

Definition 2 (maximality). *A motif x is* maximal *if any other motif y such that x occurs in y satisfies $\mathcal{L}_y \ne \mathcal{L}_x + d$ for some integer d.*

Making a maximal motif x more specific (thus obtaining y) reduces the number of its occurrences in s. Definition 2 is equivalent to that in [7] stating that x is maximal if there exist no other motif y and no integer $d \geq 0$ verifying $\mathcal{L}_x = \mathcal{L}_y + d$, such that $y[j+d] \preceq x[j]$ for $0 \leq j \leq |x| - 1$.

Definition 3 (irredundant motif). *A maximal motif x is* irredundant *if, for any maximal motifs $y_1, y_2, \ldots, y_k$ such that $\mathcal{L}_x = \cup_{i=1}^{k} \mathcal{L}_{y_i}$, motif x must be one of the y_i's. Vice versa, if all the y_i's are different from x, pattern x is said to be* covered *by motifs $y_i, y_2, \ldots, y_k$.*

The *basis of irredundant motifs* for string s is the set of *all* irredundant motifs in s, useful as a generator for all maximal motifs in s (see [7]). The size of the basis is the number of irredundant motifs contained in it. We now show the existence of an infinite family of strings s_k ($k \geq 5$) for which there are $\Omega(n^2)$ irredundant motifs in the basis already for quorum $q = 2$, where $n = |s_k|$. In this way, we disprove the upper bound of $3n$ which is based on an incorrect lemma (see also [11]). Each string s_k is the suitable extension of $t_k = \mathtt{A}^k\mathtt{T}\mathtt{A}^k$, where $\mathtt{A}^k$ denotes the letter $\mathtt{A}$ repeated k times (our argument works also for $z^k w z^k$, where $|z| = |w|$ and z is a string not sharing any common character with w). String t_k has an exponential number of maximal motifs, including those having the form $\mathtt{A}\{\mathtt{A}, \circ\}^{k-2}\mathtt{A}$ with exactly two don't cares. To see why, each such motif x occurs four times in t_k: specifically, two occurrences of x match the first and the last k letters in t_k while each distinct don't care in x matching the letter $\mathtt{T}$ in t_k contributes to one of the two remaining occurrences. Extending x or replacing a don't care with a solid character reduces the number of these occurrences, so x is maximal. The idea of our proof is to obtain strings s_k by prefixing t_k with $O(|t_k|)$ symbols to transform the above maximal motifs x into irredundant motifs for s_k. Since there are $\Theta(k^2)$ of them, and $n = |s_k| = O(|t_k|) = O(k)$, this leads to the result. In order to define s_k on the alphabet $\{\mathtt{A}, \mathtt{T}, \mathtt{u}, \mathtt{v}, \mathtt{w}, \mathtt{x}, \mathtt{y}, \mathtt{z}, \mathtt{a}_1, \mathtt{a}_2, \ldots, \mathtt{a}_{k-2}\}$, we introduce a few notations. Let $\widetilde{u}$ be the reversal of u, and let ev_k, od_k, u_k, v_k be

$$\text{if } k \text{ is even}: \begin{aligned} ev_k &= \mathtt{a}_2\mathtt{a}_4\cdots\mathtt{a}_{k-2}, \\ od_k &= \mathtt{a}_1\mathtt{a}_3\cdots\mathtt{a}_{k-3}, \\ u_k &= ev_k\,\mathtt{u}\,\widetilde{ev_k}\,\mathtt{vw}\,ev_k, \\ v_k &= od_k\,\mathtt{xy}\,\widetilde{od_k}\,\mathtt{z}\,od_k, \end{aligned} \qquad \text{if } k \text{ is odd}: \begin{aligned} ev_k &= \mathtt{a}_2\mathtt{a}_4\cdots\mathtt{a}_{k-3}, \\ od_k &= \mathtt{a}_1\mathtt{a}_3\cdots\mathtt{a}_{k-2}, \\ u_k &= ev_k\,\mathtt{uv}\,\widetilde{ev_k}\,\mathtt{wx}\,ev_k, \\ v_k &= od_k\,\mathtt{y}\,\widetilde{od_k}\,\mathtt{z}\,od_k. \end{aligned}$$

The strings s_k are then defined by $s_k = u_k v_k t_k$ for $k \geq 5$.

Lemma 1. *The length of $u_k v_k$ is $3k$, and that of s_k is $n = 5k + 1$.*

Proposition 1. *For $1 \leq p \leq k-2$, any motif of the form $\mathtt{A}^p \circ \mathtt{A}^{k-p-1}$ with one don't care cannot be maximal in s_k. Also motif $\mathtt{A}^k$ cannot be maximal in s_k.*

Proposition 2. *Each motif of the form $\mathtt{A}\{\mathtt{A}, \circ\}^{k-2}\mathtt{A}$ with exactly two don't cares is irredundant in s_k.*

Theorem 1. *The basis for string s_k contains $\Omega(n^2)$ irredundant motifs, where $n = |s_k|$ and $k \geq 5$.*

3 Tiling Motifs: The Basis and its Properties

In this section we introduce a natural notion of basis for generating all maximal motifs occurring in a string s of length n. Analogously to what was done for maximal motifs in Definition 2, we introduce displacements while defining tiling motifs for this purpose.

Definition 4 (tiling motif). *A maximal motif x is* tiling *if, for any maximal motifs $y_1, y_2, \ldots, y_k$ and for any integers $d_1, d_2, \ldots, d_k$ such that $\mathcal{L}_x = \cup_{i=1}^{k}(\mathcal{L}_{y_i} + d_i)$, motif x must be one of the y_i's. Vice versa, if all the y_i's are different from x, pattern x is said to be* tiled *by motifs $y_1, y_2, \ldots, y_k$.*

The notion of tiling is more selective than that of irredundancy in general. For example, in the string $s = \texttt{FABCXFADCYZEADCEADC}$, motif $x_1 = \texttt{A}\circ\texttt{C}$ is irredundant but it is tiled by $x_2 = \texttt{FA}\circ\texttt{C}$ and $x_3 = \texttt{ADC}$ according to Definition 4 since its location list, $\mathcal{L}_{x_1} = \{1, 6, 12, 16\}$, can be obtained from the union of $\mathcal{L}_{x_2} = \{0, 5\}$ and $\mathcal{L}_{x_3} = \{6, 12, 16\}$ with respective displacements $d_2 = 1$ and $d_3 = 0$. A fairly direct consequence of Definition 4 is that if x is tiled by $y_1, y_2, \ldots, y_k$ with associated displacements $d_1, d_2, \ldots, d_k$, then x occurs at position d_i in each y_i for $1 \leq i \leq k$ (hence $d_i \geq 0$). Note that the y_i's in Definition 4 are not necessarily distinct and that $k > 1$ for tiled motifs (it follows from the fact that $\mathcal{L}_x = \mathcal{L}_{y_1} + d_1$ with $x \neq y_1$ would contradict the maximality of both x and y_1). As a result, a maximal motif x occurring exactly q times in s is tiling as it cannot be tiled by any other motifs (we need at least two of them, which is impossible). The *basis of tiling motifs* is the complete set of all tiling motifs for s, and the size of the basis is the number of these motifs. For example, the basis $\mathcal{B}$ for $\texttt{FABCXFADCYZEADCEADC}$ contains $\texttt{FA}\circ\texttt{C}$, $\texttt{EADC}$, and $\texttt{ADC}$ as tiling motifs. Although Definition 4 is derived from that of irredundant motifs given in Definition 3, the difference is much more substantial than it may appear. The basis of tiling motifs relies on the fact that tiling motifs are considered as invariant by displacement as for maximality. Consequently, our definition of basis is symmetric, that is, each tiling motif in the basis for the reverse string $\widetilde{s}$ is the reverse of a tiling motif in the basis of s. This follows from the symmetry in Definition 4 and from the fact that maximality is also symmetric in Definition 2. It is a *sine qua non* condition for having a notion of basis invariant by the left-to-right or right-to-left order of the symbols in s (like the entropy of s), while this property does not hold for the irredundant motifs. The basis of tiling motifs has further interesting properties. Later in this section, we show that our basis is linear for quorum $q = 2$ (i.e., its size is at most $n-1$) and that the total size of the location lists for the tiling motifs is less than $2n$, describing how to find the basis in $O(n^2 \log n \log |\Sigma|)$ time. In the full paper [11], we discuss some applications such as generating all maximal motifs with the basis and finding motifs with a constraint on the number of don't cares.

Given a string s of length n, let $\mathcal{B}$ denote its basis of tiling motifs for quorum $q = 2$. Although the number of maximal motifs may be exponential and the basis of irredundant motifs may be at least quadratic (see Section 2), we show that the size of $\mathcal{B}$ is always less than n. For this, we introduce an operator $\oplus$ between the symbols of Σ to define *merges*, which are at the heart of the properties on $\mathcal{B}$. Given two letters $\sigma_1, \sigma_2 \in \Sigma$ with $\sigma_1 \neq \sigma_2$, the operator satisfies $\sigma_1 \oplus \sigma_2 = \circ$ and $\sigma_1 \oplus \sigma_1 = \sigma_1$. The operator applies to any pair of strings $x, y \in \Sigma^*$, so that $u = x \oplus y$ satisfies $u[j] = x[j] \oplus y[j]$ for all integers j. A merge is the motif resulting from applying the operator $\oplus$ to s and to its suffix at position k.

Definition 5 (Merge). *For $1 \leq k \leq n-1$, let s_k be the string whose character at position i is $s_k[i] = s[i] \oplus s[i+k]$. If s_k contains at least one solid character, $Merge_k$ denotes the motif obtained by removing all the leading and trailing don't cares in s_k (i.e., those appearing before the leftmost solid character and after the rightmost solid character).*

For example, FABCXFADCYZEADCEADC has $Merge_4 =$ EADC, $Merge_5 =$ FA∘C, $Merge_6 = Merge_{10} =$ ADC and $Merge_{11} = Merge_{15} =$ A∘C. The latter is the only merge that is not a tiling motif.

Lemma 2. *If $Merge_k$ exists, it must be a maximal motif.*

Lemma 3. *For each tiling motif x in the basis $\mathcal{B}$, there is at least one k for which $Merge_k = x$.*

Theorem 2. *Given a string s of length n and the quorum $q = 2$, let $\mathcal{M}$ be the set of $Merge_k$, for $1 \leq k \leq n-1$ such that $Merge_k$ exists. The basis $\mathcal{B}$ of tiling motifs for s satisfies $\mathcal{B} \subseteq \mathcal{M}$, and therefore the size of $\mathcal{B}$ is at most $n-1$.*

A simple consequence of Theorem 2 implies a tight bound on the number of tiling motifs for periodic strings. If $s = w^e$ for a string w repeated $e > 1$ times, then s has at most $|w|$ tiling motifs.

Corollary 1. *The number of tiling motifs for s is $\leq p$, the smallest period of s.*

The bound in Corollary 1 is not valid for irredundant motifs. String $s =$ ATATATATA has period $p = 2$ and only one tiling motif ATATATA, while its irredundant motifs are A, ATA, ATATA and ATATATA.

We describe how to compute the basis $\mathcal{B}$ for string s when $q = 2$. A brute-force algorithm generating first all maximal motifs of s takes exponential time in the worst case. Theorem 2 plays a crucial role in that we first compute the motifs in $\mathcal{M}$ and then discard those being tiled. Since $\mathcal{B} \subseteq \mathcal{M}$, what remains is exactly $\mathcal{B}$. To appreciate this approach, it is worth noting that we are left with the problem of selecting $\mathcal{B}$ from $n-1$ maximal motifs in $\mathcal{M}$ at most, rather than selecting $\mathcal{B}$ among *all* the maximal motifs in s, which may be exponential in number. Our simple algorithm takes $O(n^2 \log n \log |\Sigma|)$ time and is faster than previous (and more complicated) methods.

Step 1. Compute the multiset $\mathcal{M}'$ of merges. Letting $s_k[i]$ be the leftmost solid character of string s_k in Definition 5, we define $occ_x = \{i, i+k\}$ to be the positions of the two occurrences of x whose superposition generates $x = Merge_k$. For $k = 1, 2, \ldots, n-1$, we compute string s_k in $O(n-k)$ time. If s_k contains some solid characters, we compute $x = Merge_k$ and occ_x in the same time complexity. As a result, we compute the multiset $\mathcal{M}'$ of merges in $O(n^2)$ time. Each merge x in $\mathcal{M}'$ is identified by a triplet $\langle i, i+k, |x| \rangle$, from which we can recover the jth symbol of x in constant time by simple arithmetic operations and comparisons.

Step 2. Transform the multiset $\mathcal{M}'$ into the set $\mathcal{M}$ of merges. Since there can be two or more merges in $\mathcal{M}'$ that are identical and correspond to the same merge in $\mathcal{M}$, we put together all identical merges in $\mathcal{M}'$ by performing radix sorting on the triplets representing them. The total cost of this step is dominated by radix sorting, giving $O(n^2 \log |\Sigma|)$ time. As byproduct, we produce the temporary location list $T_x = \bigcup_{x'=x\,:\,x' \in \mathcal{M}'} occ_{x'}$ for each distinct $x \in \mathcal{M}$ thus obtained.

Lemma 4. *Each motif $x \in \mathcal{B}$ satisfies $T_x = \mathcal{L}_x$.*

Step 3. Select $\mathcal{M}^* \subseteq \mathcal{M}$, where $\mathcal{M}^* = \{x \in \mathcal{M} : T_x = \mathcal{L}_x\}$. In order to build $\mathcal{M}^*$, we employ the Fischer-Paterson algorithm based on convolution [5] for string matching with don't cares to compute the whole list of occurrences $\mathcal{L}_x$ for each merge $x \in \mathcal{M}$. Its cost is $O((|x| + n) \log n \log |\Sigma|)$ time for each merge x. Since $|x| < n$ and there are at most $n-1$ motifs $x \in \mathcal{M}$, we obtain $O(n^2 \log n \log |\Sigma|)$ time to construct all lists $\mathcal{L}_x$. We can compute $\mathcal{M}^*$ by discarding the merges $x \in \mathcal{M}$ such that $T_x \neq \mathcal{L}_x$ in additional $O(n^2)$ time.

Lemma 5. *The set $\mathcal{M}^*$ satisfy the conditions $\mathcal{B} \subseteq \mathcal{M}^*$ and $\sum_{x \in \mathcal{M}^*} |\mathcal{L}_x| < 2n$.*

The property of $\mathcal{M}^*$ in Lemma 5 is crucial in that $\sum_{x \in \mathcal{M}} |\mathcal{L}_x| = \Theta(n^2)$ when many lists contain $\Theta(n)$ entries. For example, $s = \mathtt{A}^n$ has $n-1$ distinct merges, each of the form $x = \mathtt{A}^i$ for $1 \leq i \leq n-1$, and so $|\mathcal{L}_x| = n - i + 1$. This would be a sharp drawback in Step 4 when removing tiled motifs as it may turn into an $\Theta(n^3)$ algorithm. Using $\mathcal{M}^*$ instead, we are guaranteed that $\sum_{x \in \mathcal{M}^*} |\mathcal{L}_x| = O(n)$; we may still have some tiled motifs in $\mathcal{M}^*$, but their total number of occurrences is $O(n)$.

Step 4. Discard the tiled motifs in $\mathcal{M}^*$. We can now check for tiling motifs in $O(n^2)$ time. Given two distinct motifs $x, y \in \mathcal{M}^*$, we want to test whether $\mathcal{L}_x + d \subseteq \mathcal{L}_y$ for some integer d and, in that case, we want to mark the entries in $\mathcal{L}_y$ that are also in $\mathcal{L}_x + d$. At the end of this task, the lists having *all* entries marked are tiled (see Definition 4). By removing their corresponding motifs from $\mathcal{M}^*$, we eventually obtain the basis $\mathcal{B}$ by Lemma 5. Since the meaningful values of d are equal to the individual entries of $\mathcal{L}_y$, we have only $|\mathcal{L}_y|$ possible values to check. For a *given* value of d, we avoid to merge $\mathcal{L}_x$ and $\mathcal{L}_y$ in $O(|\mathcal{L}_x| + |\mathcal{L}_y|)$ time to perform the test, as it would contribute to a total of $\Theta(n^3)$ time. Instead, we exploit the fact that each list has values ranging from 1 to n, and use a couple of bit-vectors of size n to perform the above check in $O(|\mathcal{L}_x| \times |\mathcal{L}_y|)$ time for *all* values of d. This gives $O(\sum_y \sum_x |\mathcal{L}_x| \times |\mathcal{L}_y|) = O(\sum_y |\mathcal{L}_y| \times \sum_x |\mathcal{L}_x|) = O(n^2)$ by Lemma 5. We therefore detail how to perform the above check with $\mathcal{L}_x$ and $\mathcal{L}_y$ in $O(|\mathcal{L}_x| \times |\mathcal{L}_y|)$ time. We use two bit-vectors V_1 and V_2 initially set to all zeros. Given $y \in \mathcal{M}^*$, we set $V_1[i] = 1$ if $i \in \mathcal{L}_y$. For each $x \in \mathcal{M}^* - \{y\}$ and for each $d \in \mathcal{L}_y$, we then perform the following test. If *all* $j \in \mathcal{L}_x + d$ satisfy $V_1[j] = 1$, we set $V_2[j] = 1$ for all such j. Otherwise, we take the next value of d, or the next motif if there are no more values of d, and we repeat the test. After examining all $x \in \mathcal{M}^* - \{y\}$, we check whether $V_1[i] = V_2[i]$ for all $i \in \mathcal{L}_y$. If so,

y is tiled as its list is covered by possibly shifted location lists of other motifs. We then reset the ones in both vectors in $O(|\mathcal{L}_y|)$ time.

Summing up Steps 1–4, the dominant cost is that of Step 3, leading to the following result.

Theorem 3. *Given an input string s of length n over the alphabet Σ, the basis of tiling motifs with quorum $q = 2$ can be computed in $O(n^2 \log n \log |\Sigma|)$ time. The total number of motifs in the basis is less than n, and the total number of their occurrences in s is less than $2n$.*

4 $q > 2$: Pseudo-Polynomial Bases for Higher Quorum

We now discuss the general case of quorum $q \geq 2$ for finding the basis of a string of length n. Differently from previous work claiming a polynomial-time algorithm for *any* arbitrary value of q, we show in Section 4 that *no* such polynomial-time algorithm can exist in the worst case, both for the basis of irredundant motifs and for the basis of tiling motifs. The size of these bases provably depends *exponentially* on suitable values of $q \geq 2$, i.e., we give a lower bound of $\Omega\binom{\frac{n-1}{2}-1}{q-1}$. In practice, this size has an exponential growth for increasing values of q up to $O(\log n)$, but larger values of q are theoretically possible in the worst case. Fixing $q = (n-1)/4 + 1$ in our lower bound, we get a size of $\Omega(2^{(n-1)/4})$ motifs in the bases. On the average $q = O(\log_{|\Sigma|} n)$ by extending the argument after Theorem 3. We show a further property for the basis of tiling motifs in Section 4, giving an upper bound of $\binom{n-1}{q-1}$ on its size with a simple proof. Since we can find an algorithm taking time proportional to the square of that size, we can conclude that a polynomial-time algorithm for finding the basis of tiling motifs exists in the worst case if and only if the quorum q satisfies either $q = O(1)$ or $q = n - O(1)$ (the latter condition is hardly meaningful in practice).

We now show the existence of a family of strings for which there are at least $\binom{\frac{n-1}{2}-1}{q-1}$ tiling motifs for a quorum q. Since a tiling motif is also irredundant, this gives a lower bound for the irredundant motifs to be combined with that in Section 2 (the latter lower bound still gives $\Omega(n^2)$ for $q \geq 2$). The strings are this time $t_k = \mathtt{A}^k\mathtt{T}\mathtt{A}^k$ $(k \geq 5)$ themselves, without the left extension used in the bound of Section 2. The proof proceeds by exhibiting $\binom{k-1}{q-1}$ motifs that are maximal and have each exactly q occurrences, from whence it follows immediately that they are tiling (indeed the remark made after Definition 4 holds for any $q \geq 2$).

Proposition 3. *For $2 \leq q \leq k$ and $1 \leq p \leq k-q+1$, any motif $\mathtt{A}^p \circ \{\mathtt{A}, \circ\}^{k-p-1} \circ \mathtt{A}^p$ with exactly q don't cares is tiling (and so irredundant) in t_k.*

Theorem 4. *String t_k has $\binom{\frac{n-1}{2}-1}{q-1} = \Omega\left(\frac{1}{2^q}\binom{n-1}{q-1}\right)$ tiling (and irredundant) motifs, where $n = |t_k|$ and $k \geq 2$.*

We now prove that $\binom{n-1}{q-1}$ is, instead, an upper bound for the size of a basis of tiling motifs for a string s and quorum $q \geq 2$. Let us denote as before such a basis by $\mathcal{B}$. To prove the upper bound, we use again the notion of a merge except that it involves q strings. The operator $\oplus$ between the elements of Σ is the same as before. Let k be an array of $q-1$ positive values $k_1, \ldots, k_{q-1}$ with $1 \leq k_i < k_j \leq n-1$ for all $1 \leq i < j \leq q-1$. A merge is the (non empty) pattern that results from applying the operator $\oplus$ to the string s and to s itself $q-1$ times, at each time shifted by k_i positions to the right for $1 \leq i \leq q-1$.

Lemma 6. *If $Merge_k$ exists for quorum q, it must be a maximal motif.*

Lemma 7. *For each tiling motif x in the basis $\mathcal{B}$ with quorum q, there is at least one k for which $Merge_k = x$.*

Theorem 5. *Given a string s of length n and a quorum q, let $\mathcal{M}$ be the set of $Merge_k$, for any of the $\binom{n-1}{q-1}$ possible choices of k for which $Merge_k$ exists. The basis $\mathcal{B}$ of tiling motifs satisfies $\mathcal{B} \subseteq \mathcal{M}$, and therefore $|\mathcal{B}| \leq \binom{n-1}{q-1}$.*

The tiling motifs in our basis appear in s for a total of $q\binom{n-1}{q-1}$ times at most. A generalization of the algorithm given in Section 3 gives a pseudo-polynomial time complexity of $O\left(q^2\binom{n-1}{q-1}^2\right)$.

References

1. A. Apostolico. Pattern discovery and the algorithmics of surprise. In *NATO ASI on Artificial Intelligence and Heuristic Methods for Bioinformatics.* IOS press, 2003.
2. A. Apostolico. Personal communication, May 2003.
3. A. Apostolico and L. Parida. Incremental paradigms of motif discovery. unpublished, 2002.
4. A. Apostolico and L. Parida. Compression and the wheel of fortune. In *IEEE Data Compression Conference (DCC'2003)*, pages 143–152, 2003.
5. M. Fischer and M. Paterson. String matching and other products. In R. Karp, editor, *SIAM AMS Complexity of Computation*, pages 113–125, 1974.
6. H. Mannila. Local and global methods in data mining: basic techniques and open problems. In P. et al., editor, *International Colloquium on Automata, Languages, and Programming*, volume 2380 of *LNCS*, pages 57–68. Springer-Verlag, 2002.
7. L. Parida, I. Rigoutsos, A. Floratos, D. Platt, and Y. Gao. Pattern Discovery on Character Sets and Real-valued Data: Linear Bound on Irredundant Motifs and Efficient Polynomial Time Algorithm. In *SIAM Symposium on Discrete Algorithms*, 2000.
8. J. Pelfrêne, S. Abdeddaïm, and J. Alexandre. Un algorithme d'indexation de motifs approchés. In *Journée Ouvertes Biologie Informatique Mathématiques (JOBIM)*, pages 263–264, 2002.
9. J. Pelfrêne, S. Abdeddaïm, and J. Alexandre. Extracting approximare patterns. In *Combinatorial Pattern Matching*, 2003. to appear.
10. N. Pisanti, M. Crochemore, R. Grossi, and M.-F. Sagot. A basis for repeated motifs in pattern discovery and text mining. Technical Report IGM 2002-10, Institut Gaspard-Monge, University of Marne-la-Vallée, July 2002.
11. N. Pisanti, M. Crochemore, R. Grossi, and M.-F. Sagot. Bases of motifs for generating repeated patterns with don't cares. Technical Report TR-03-02, Dipartimento di Informatica, University of Pisa, January 2003.