Characterization and Extraction of Irredundant Tandem Motifs

Laxmi Parida[1], Cinzia Pizzi[2], and Simona E. Rombo[3]

[1] IBM T.J. Watson Research Center
[2] Department of Information Engineering, University of Padova
[3] ICAR-CNR of Cosenza & DEIS, Università della Calabria

Abstract. We address the problem of extracting pairs of subwords (m_1, m_2) from a text string s of length n, such that, given also an integer constant d in input, m_1 and m_2 occur in tandem within a maximum distance of d symbols in s.

The main effort of this work is to eliminate the possible redundancy from the candidate set of the so found *tandem motifs*. To this aim, we first introduce the concept of *maximality*, characterized by four specific conditions, that we show to be not deducible by the corresponding notion of maximality already defined for "simple" (i.e., non tandem) motifs. Then, we further eliminate the remaining redundancy by defining the concept of *irredundancy* for tandem motifs.

We prove that the number of non-overlapping irredundant tandems is $O(d^2n)$ which, considering d as a constant, leads to a linear number of tandems in the length of the input string. This is an order of magnitude less than previously developed compact indexes for tandem extraction. As a further contribution we show an algorithm to extract this compact irredundant index.

1 Introduction

Extracting pairs (or sets) of subwords that often occur together in an input string is an important task in different application contexts, such as for example bioinformatics [15,14] or natural language processing [3]. In the last few years, several approaches have been proposed (e.g., [8,9,12,13]) dealing with the most general version of the problem, that is, extracting sets of subwords that occur (also non exactly) together in a given sequence, within a distance that is fixed in a finite range. Despite of their flexibility, such techniques do not care of avoiding redundancy in the output solutions, that can become also very large, especially when the input string is much repetitive. For the case of solid components in [3] a compact index was proposed to compute the number of co-occurrence within a given distance of any pair of substrings of an input string, without interleaving occurrences, in time and space quadratic in the length of the input. In [5] this bound was improved to the actual size of the output. In [4] distances other than beginning-to-beginning were considered. However, these works on compact indexes considered tandems between (right-)maximal components, and did not take into consideration the maximality of the tandem itself.

L. Calderón-Benavides et al. (Eds.): SPIRE 2012, LNCS 7608, pp. 385–397, 2012.

Our approach addresses the problem of extracting pairs of subwords (m_1, m_2) from a text string s of length n, such that, given also two integer constants d and q in input, m_1 and m_2 occur in tandem at least q times within a maximum distance of d symbols (from the beginning of each component) in s. We call *tandem motifs* such repeated subword pairs[1]. Differently from previous work, we aim at eliminating *all* the redundancy that can be implicitly contained in the output generation. In particular, we define a new class of tandem motifs, that we called *irredundant tandem motifs*, able to represent in a compact way all the possible tandem motifs that can be extracted from s. We show that irredundant tandem motifs cannot be trivially obtained by the companion notions of maximality and irredundancy already studied for motifs without co-occurrences (see, e.g., [2,6,7,10,11,16,17,18,19,20,21]).

Note that tandem motifs as defined in this paper can be also related to the notion of generalized extensible motifs addressed in [1]. However, tandem motifs are particularly interesting because of their additional properties (shown in this paper) that do not hold for the generalized extensible motifs. Furthermore, the class of extensible motifs can contain some redundancy, differently from the class of tandem motifs we propose in this work.

The paper is organized as follows. In Section 2 we introduce some preliminary definitions and some properties that are important for the rest of the analysis. In Section 3 we show some bounds on the number of tandem motifs that can be extracted from a string. Section 4 presents a procedure to extract irredundant tandem motifs. Finally, in Section 5 we draw our conclusive remarks.

2 Properties and Definitions

We now introduce some suitable definitions needed for the formalization of the problem.

In the following, given in input a string s of n characters on the alphabet Σ, we denote by $s[i]$ the i-th element in s. Furthermore, we denote by $|X|$ the size of a set X, and by $|y|$ the length of a subword y. Given two strings y_1 and y_2 (e.g., two subwords of s), y_1y_2 indicates the concatenation of y_1 and y_2.

Definition 1. *(Exact occurrence)* A string s' of size n' $(n' \leq n)$ *occurs exactly* at the position h in s $(h \leq n - n')$ if $s[i + h - 1] = s'[i]$, for each $i = 1 \dots n'$.

Definition 2. *(Substring)* A string $s' = s'_1 \dots s'_{n'}$ $(n' \leq n)$ is a *substring* of s if there exists a position h of s $(h \leq n - n')$ such that s' occurs exactly at h in s.

[1] In [3,5] the notion of *tandem* implies that between two substrings there are no interleaving occurrences of one or the other. Here we do not impose such a constraint. In [8,12,13] the term *structured motif* refer to a similar kind of motif. However, the distance between components is measured differently. For this reason we prefer to use the term *tandem motif* as in [3,5] where the distance was measured the same way as in this present work. It is also worth noting, to avoid confusion, that *tandem motifs* are unrelated to *tandem repeats*.

Definition 3. *(Tandem, Occurrence)* Let d be a positive integer (aka *distance*) such that $d \leq n$, and m_1 and m_2 be two substrings of s. The pair $t = \langle m_1, m_2 \rangle$ is a *tandem* with *components* m_1 and m_2 if there exist two positions i and j of s such that $1 \leq j - i \leq d$ and m_1 and m_2 occur exactly at i and j, respectively. In this case we say that the tandem t occurs at $\ell = (i, j)$ in s.

Note that taking as distance d the number of characters between the beginning of the first component and the beginning of the second component allows to easily intercept also tandem occurrences where the two components overlap. However, special cases such as tandems whose components never overlap can be managed as well, as will be discussed later in the paper.

Definition 4. *(Sub-tandem)* Let $t' = \langle m'_1, m'_2 \rangle$ and $t'' = \langle m''_1, m''_2 \rangle$ be two tandems w.r.t. the same distance d. The tandem t' is a *sub-tandem* of t'' ($t' \preceq t''$) if and only if m'_1 and m'_2 are substrings of m''_1 and m''_2, respectively.

Definition 5. *(Tandem q-motif, Location list)* Let q be a positive integer (aka *quorum*) such that $q \leq n$, and $t = \langle m_1, m_2 \rangle$ be a *tandem*. The tandem t is a *tandem q-motif* of s with *location list* $\mathcal{L}_t = \{\ell_1, \ell_2, \ldots, \ell_p\}$, if all the following hold:

1. t occurs at ℓ_i for each $\ell_i \in \mathcal{L}_t$;
2. $p \geq q$;
3. there is no pair $\ell \neq \ell_h$, $1 \leq h \leq p$ such that t occurs at ℓ in s (the location list is of maximal size).

Whenever the value of q is clear from the context, we call a tandem q-motif *tandem motif*. In this paper, we focus on the case of $q = 2$.

Definition 6. (Maximal tandem motif) A tandem motif $t = \langle m_1, m_2 \rangle$ with location list $\mathcal{L}_t$ is *maximal* if and only if there is no tandem motif $t' = \langle m'_1, m'_2 \rangle$ with location list $\mathcal{L}_{t'}$ such that both m_1 and m_2 are equal to or are substrings of m'_1 and m'_2, respectively, and $|\mathcal{L}_t| = |\mathcal{L}_{t'}|$.

Due to the composite nature of a tandem motif, both the components concur to its maximality. A first question is if there is some relation between the maximality of each of the two components[2] and the maximality of the corresponding tandem motif. Intuitively, the maximality of a tandem motif cannot be deduced by the (possible) maximality of its components, as shown by the example below. For maximality of a component m we mean that there does not exist any substring m' in s such that m is a substring of m' and the number of occurrences of m is equal to that of m'.

Example 1. Let:

a	**b**	**b**	**a**	**a**	d	a	**b**	**b**	c	d	**a**	**a**	b	a	**b**	**b**	a	c	**a**	**a**	b	a	c	c	c	c	c	a	a	d	a
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32

[2] See [16,18] for a formal definition of (non tandem) maximal motifs.

be the input string s and $d = 4$ be the allowed distance. Consider the tandem $t = \langle bb, aa \rangle$, occurring at $(2, 4)$, $(8, 12)$ and $(16, 20)$ in s. It is easy to see that t is maximal, although its first component is not. Indeed, bb is a substring of abb, and both bb and abb occur exactly three times in s, but $\langle abb, aa \rangle$ has only one occurrence satisfying the distance constraint $d = 4$ (i.e., $(1, 4)$). Analogously, not all the maximal substrings in s necessarily concur to be part of a tandem motif: as an example, $aada$ is maximal but it is not followed by (and it does not follow) any other substring which can represent a suitable component for a candidate tandem motif.

The example above confirms that the "tandem-maximality" cannot be checked by a simple analysis of each single component alone. However, the following four different conditions allow to discriminate if a tandem motif $t = \langle m_1, m_2 \rangle$ is maximal. All these conditions have to be handled properly in order to extract maximal tandem motifs, as will be detailed in Section 4.

1. *Right maximality of the second component (RMSC).* This condition means that the second component cannot be extended by adding any character on the right without loosing some occurrence in $\mathcal{L}_t$. In other words, there is no substring m'_2 such that $|m_2| < |m'_2|$, $m_2[i] = m'_2[i]$ for $i = 1, \ldots, |m_2|$ and $|\mathcal{L}_t| = |\mathcal{L}_{t'}|$ if $t' = \langle m_1, m'_2 \rangle$.
2. *Right maximality of the first component (RMFC).* In this case, the first component cannot be extended by adding any character on the right without loosing some occurrence in $\mathcal{L}_t$. Thus, there is no substring m'_1 such that $|m_1| < |m'_1|$, $m_1[i] = m'_1[i]$ for $i = 1, \ldots, |m_1|$ and $|\mathcal{L}_t| = |\mathcal{L}_{t'}|$ if $t' = \langle m'_1, m_2 \rangle$.
3. *Left tandem maximality of the second component (LMSC).* The second component is *left maximal* if there is no substring m'_2 such that $|m_2| < |m'_2|$, $m_2[i] = m'_2[i + h]$ for $i = 1, \ldots, |m_2|$ where $h = |m'2| - |m_2|$, $t' = \langle m_1, m'_2 \rangle$ is a tandem motif, and $|\mathcal{L}_t| = |\mathcal{L}_{t'}|$.
4. *Left tandem maximality of the first component (LMFC).* The first component is *left maximal* if there are no substrings m'_1 and m'_2 such that: *(i)* $|m_1| < |m'_1|$ and $|m_2| \leq |m'_2|$, *(ii)* $m_1[i] = m'_1[i + h]$ for $i = 1, \ldots, |m_1|$, *(iii)* $h = |m'_1| - |m_1|$, *(iv)* $m'_2 = x \cdot m_2$, where the symbol $\cdot$ represents the concatenation between strings, and x is a substring in s such that $t' = \langle m'_1, m'_2 \rangle$ is a tandem motif (i.e., the constraint on the distance between m'_1 and m'_2 is satisfied; note that x can also coincide with the empty string), and *(v)* $|\mathcal{L}_t| = |\mathcal{L}_{t'}|$.

We anticipate that, as better pointed out in Section 4, LMFC has operatively to be handled after LMSC. Figure 1 shows an example for each type of maximality. Although the concept of maximality allows us to consistently reduce the size of the output set, without any information loss, there is still some residual redundancy that is related to the occurrences of the tandem motifs, rather than to their structural composition. Indeed, different maximal tandem motifs could cover overlapping regions of the input string.

Let $t = \langle m_1, m_2 \rangle$ be a tandem motif with location list $\mathcal{L} = \{(i_1, j_1), \ldots, (i_p, j_p)\}$ and f and g be two shifting integers. We call *shifted location list*

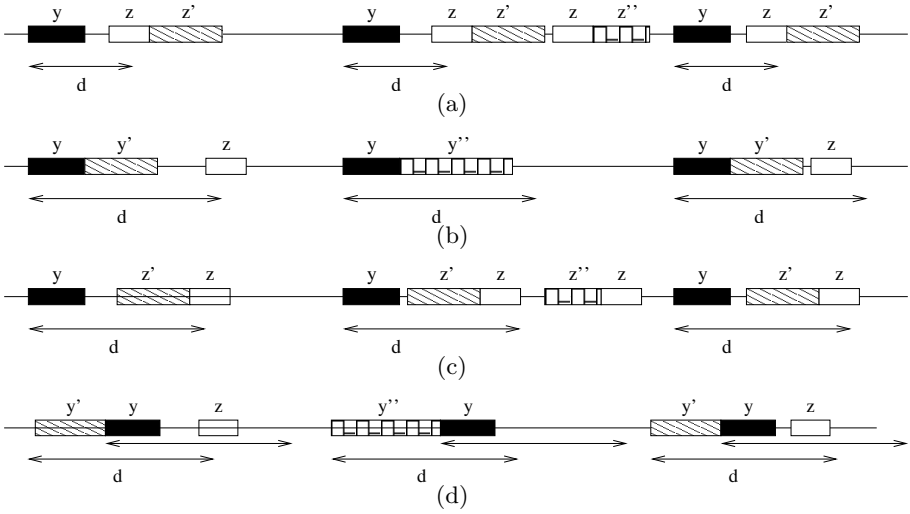

Fig. 1. Examples of maximalities: (a) Right-maximality of the second component z: whenever an occurrence of the second component z falls within distance d from an occurrence of y to its left, z is always followed by a string z'. The maximal tandem is $\langle y, zz' \rangle$. (b) Right-maximality of the first component y: whenever an occurrence of the second component z falls within distance d from an occurrence of y to its left, y is always followed by a string y'. The maximal tandem is $\langle yy', z \rangle$. (c) Left-maximality of the second component z: whenever an occurrence of the second component z falls within distance d from an occurrence of y to its left, z is always preceded by a string z'. The distance between y and z must be always positive to be valid. The maximal tandem is $\langle y, z'z \rangle$. (d) Left-maximality of the first component y: whenever an occurrence of the second component z falls within distance d from an occurrence of y to its left, y is always preceded by a string y'. The distance between y' and z is always less than or equal to d. The maximal tandem is $\langle y'y, z \rangle$.

$\mathcal{L}^s = \{(i_1 + f, j_1 + g), \ldots, (i_p + f, j_p + g)\}$ the list of locations obtained by adding f to each occurrence of the first component and g to each occurrence of the second component, respectively.

The following definition is useful to discard those maximal tandem motifs that are not essential, and that can be deduced by other maximal tandem motifs.

Definition 7. (Irredundant tandem motif) A maximal tandem motif $t = \langle m_1, m_2 \rangle$ with location list $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_p\}$ is *redundant* if and only if there exist k tandem motifs $t_1, \ldots, t_k$ with location lists $\mathcal{L}_1, \ldots, \mathcal{L}_k$, respectively, and two sets $F = \{f_1, \ldots, f_k\}$ and $G = \{g_1, \ldots, g_k\}$ of proper shifting integers such that:

- t is a sub-tandem of each t_h $(1 \leq h \leq k)$,
- $\mathcal{L} = \{\mathcal{L}_1^s \cup \ldots \cup \mathcal{L}_k^s\}$, where each $\mathcal{L}_h^s$ is the shifted location list of t_h obtained by exploiting f_h and g_h as shifting integers.

A maximal tandem motif that is not redundant is called *irredundant*.

Example 2. Let:

a b b a a d a b b c d a a a a b b a c a a a a c b b c a a d a

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31

be the input string and $d = 5$ be the allowed distance. Consider the three maximal tandem motifs $t' = \langle bb, aa \rangle$, $t'' = \langle abb, aaaa \rangle$, $t''' = \langle bb, aada \rangle$, with location lists $\mathcal{L}' = \{(2,4),(8,12),(16,20),(25,28)\}$, $\mathcal{L}'' = \{(7,12),(15,20)\}$ and $\mathcal{L}''' = \{(2,4),(25,28)\}$, respectively. Then, $\mathcal{L}' = \{(7+1,12),(15+1,20)\} \cup \{(2,4),(25,28)\}$, that is, there exist $F = \{1,0\}$ and $G = \{0,0\}$ such that the definition above is satisfied thus t' is redundant. It is easy to see that both t'' and t''' are irredundant.

Definition 8. *(Exposed occurrence)* A position (i,j) of s is an *exposed occurrence* of the maximal tandem motif t' if t' occurs at (i,j), and there does not exist any other maximal tandem motif t'' such that $t' \preceq t''$ and t'' occurs at $(i-f, j-g)$ $(f,g \geq 0)$, with f and g proper shifting integers.

3 The Number of Tandem Motifs

We now discuss the size of the special classes of motifs that we introduced in this work. An important problem is to understand how many irredundant tandem motifs can be extracted from the input string. To this aim, it is worth to point out that the set of irredundant motifs is contained in the set of maximal motifs, by Definition 7.

Theorem 1. A maximal tandem t of a string s is irredundant if and only if it has at least one exposed occurrence in s.

Proof. Let (i,j) be an exposed occurrence of t, and suppose that t is redundant. Then, there exist k tandem motifs $t_1, \ldots, t_k$ such that t is a sub-tandem of each t_h $(1 \leq h \leq k)$ and its occurrence list $\mathcal{L}$ is equal to the union of their occurrence lists, unless some displacements. This means that *each* occurrence of t has to be covered by another occurrence of some t_h $(1 \leq h \leq k)$. Thus, some t_h occurs at $(i-f, j-g)$ $(f,g \geq 0)$ and $t \preceq t_h$, that is, a contradiction.

Now we prove the converse. Suppose that t is irredundant and that no one of its occurrences is exposed. Thus, at each (i,j) where t occurs, there occurs (unless some proper displacements) also some maximal motif t' such that $t \preceq t'$. The union of the occurrences lists of all such t' gives the occurrence list of t. This means that t is redundant, that is, a contradiction. □

3.1 Number of Candidate Tandems

We recall that the number of substrings made of only solid symbols of s is $O(n^2)$, thus there are $O(n^4)$ pairs of subwords. If we consider a fixed maximum distance d, the possible pairs become $O(dn^3)$, as proved in the following lemma.

Lemma 1. Given a string s of length n and a distance $d > 0$, the number of possible pairs of solid substrings at (head-to-head) distance at most d is $O(dn^3)$.

Proof. In a string of length n there are $O(n^2)$ substrings, each of which can be followed by at most dn components. Hence the total number of tandem motifs is $O(dn^3)$.

□

Lemma 1 provides a bound for the number of possible tandem motifs that can be extracted from the input string s. In [3] the authors show also how it is possible to build an $O(n^2)$ index that stores the co-occurrences of all the pairs of strings that correspond to the node of the suffix tree (i.e right-maximal), and gives the co-occurrence count for any pair.

We now consider the special class of irredundant tandem motifs, with solid components.

3.2 Number of Irredundant Tandems

Let $t = \langle m_1, m_2 \rangle$ be a tandem motif in s. We say that m_1 and m_2 are *non-overlapping* components if there is no occurrence of t in s where m_1 and m_2 overlap. We say that m_1 and m_2 are *overlapping* components otherwise. The case of overlapping components can be reduced to the search for longer single words, for which efficient algorithms and data structures already exists, so we will focus on non-overlapping components.

Theorem 2. Let s be a string of length n on a generic alphabet Σ. Then, the number of irredundant tandem motifs $t = \langle m_1, m_2 \rangle$ in s with m_1 and m_2 non-overlapping solid components is $O(d^2 n)$.

Proof. Let $\mathcal{T}$ be the set of irredundant tandem motifs with non-overlapping solid components in s. We recall that, from Theorem 1, each $t \in \mathcal{T}$ has at least an exposed occurrence. Given a generic position (i, j) of s, we want to know the maximum number of motifs in $\mathcal{T}$ that can simultaneously have an exposed occurrence at (i, j).

Starting from position i, there are at most $d - 1$ different subwords that can concur to be the first component of some motifs in $\mathcal{T}$ without intercepting the second component, that starts at position j of s. For each of such first components, there is at most one subword starting at position j that can be the second component of a motif in $\mathcal{T}$ with an exposed occurrence at (i, j). Indeed, let $t' = \langle m'_1, m'_2 \rangle$ and $t'' = \langle m''_1, m''_2 \rangle$ be two motifs in $\mathcal{T}$ both having an exposed occurrence at (i, j), and suppose for contradiction that $m'_1 = m''_1$ but $m'_2 \neq m''_2$. Since both m'_2 and m''_2 start at position j, one between $m'_2 \preceq m''_2$ or $m''_2 \preceq m'_2$ necessarily holds. Thus, one between $t' \preceq t''$ or $t'' \preceq t'$ necessarily holds as well. This leads to a contradiction since both t' and t'' were assumed to have an exposed occurrence at (i, j).

Thus, for each position (i, j) of s there are at most $O(d)$ different motifs in $\mathcal{T}$ with an exposed occurrence at (i, j). Since the number of position (i, j) at distance at most d is $O(dn)$ the claim is proved. □

The following example clarifies Theorem 2.

Example 3. Let:

a	a	a	a	b	c	a	b	b	b	b	r	r	r	r	r	r	a	a	a	a	c	c	c	b	b	r	r	r	r	r	r	a	a	c	c	c	b	b	b	b
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41

be the input string s and $d = 7$ be the allowed distance. Consider for example the location $(1, 8)$ of s. Both the two irredundant tandem motifs $t' = \langle aaaa, bb\rangle$ and $t'' = \langle aa, bbbb\rangle$ occur at such location with an exposed occurrence.

4 Algorithms for Irredundant Tandem Extraction

This section describes a procedure to extract the set of irredundant tandems motifs in an input string s. For this purpose, we consider a variant of the tandem trees introduced in [3], and we exploit a three-steps approach: *i)* build tandem trees to capture the number of co-occurrences between substrings in s; *ii)* extend the candidate components to obtain maximal tandems; *iii)* eliminate redundancy from maximal candidates to obtain irredundant tandems.

4.1 Tandem Trees

A tandem tree D_y is a suffix tree, associated with a substring y of s, in which each node α is annotated with the co-occurrence count between y and $z = w(\alpha)$, where $w(\alpha)$ is the string spelled out by the path from the root to α.

In [3] a tandem tree is built for any substring y that has a proper locus in the suffix tree of s. In such a way the number of co-occurrences within distance d is explicitly computed only between substrings that have a proper locus in the suffix tree of s. The authors showed that this suffices to represent the number of co-occurrences between *any* substring in s. In fact for any pair (y', z') that is not explicitly indexed there is a pair (y, z), with the same number of co-occurrences, such that: i) y is the string corresponding to the locus of y'; ii) z is the string that correspond to the locus of z'; the co-occurrence count is stored in D_y, at a node α s.t. $z = w(\alpha)$.

Let $P = \{p_1 \dots p_k\}$ be the occurring positions of a string y corresponding to a proper locus in the suffix tree of s. Let $L(p)$ be a mapping from each position p and the leaf corresponding to the suffix that starts at position p. The basic steps to build the tandem tree D_y of a string y are:

1. assign to all leaves a zero weight;
2. for all $p \in P$ mark the positions $p + i$, $1 \leq i \leq d$, and for each marked position m add 1 to $L(m)$;
3. annotate the tree bottom up so that the weight of an internal node is the sum of the weights of its children.

For our purposes we first have to modify the algorithm to mark in D_y the leaves corresponding to the positions $p + |y| + i$, with $1 \leq i \leq d - |y|$, rather than $p + i$ to avoid overlaps.

Then we can safely extract maximal tandem motifs starting from this reduced $O(n^2)$ set of pairs of strings (that can be stored as $O(n)$ tandem trees D_y). Indeed, as we just discussed, the pairs of strings that are not indexed are surely not maximal since they co-occur the same number of times (and for the properties of the suffix tree with the same location list) of a pair that extends both components up to their locus in the suffix tree.

As discussed in Section 2 when considering tandem motifs we might have the possibility to further extend the components of this pair to the right and to the left. We now discuss how we handle all four kinds of extensions to perform the second step of our approach.

4.2 Right-Maximality of the Second Component

In a tandem tree this situation can be visualized when, in the annotation of an internal node, all the contributions come from a single child. The right-maximality of the second component can be obtained as explained in [5]:

- build D_y with the procedure described in 4.1;
- if node α has a null score, delete α and the subtree rooted at α;
- if there is a path of nodes with a single child, then compact the path in a unique edge.

The entire procedure can be carried out in time proportional to the size of the output [5], which is upper bounded by $O(n^2)$.

4.3 Right-Maximality of the First Component

A pair (y, z) is not right-maximal w.r.t. the first component if we can find a pair (yy', z) with the same number of co-occurrences. Since the right maximality does not change the starting positions, this equality implies the location lists must also be the same.

In order to eliminate this kind of not maximal pairs, we traverse the suffix tree T of s with a depth first visit. For each node ν, with associated string $y = w(\nu)$, we consider its children $\nu_1, \nu_2, \ldots \nu_k$ with associated strings $yy_1, yy_2, \ldots, yy_k$, and the corresponding tandem trees D_y, $D_{yy_1} \ldots D_{yy_k}$.

We will have that if the weight of some node β, with $z = w(\beta)$, in the tandem tree D_{yy_i} is the same as the weight computed for y in D_y, the tandem $\langle yy_i, z \rangle$ covers $\langle y, z \rangle$. The same will obviously hold also for any children of β, so we can safely remove this node and the subtree rooted at it from D_y. In fact, the corresponding maximal pairs are indexed in D_{yy_i}. When all the children have been considered D_y is traversed to merge paths that have been eventually left with a single child chain.

Each node plays the role of the child only when its father is chosen as ν, so the overall number of children that we consider is exactly the number of nodes

in the suffix tree, i.e $O(n)$. Since each time we traverse a tandem tree, taking $O(n)$ time, the total time complexity of this step is $O(n^2)$.

4.4 Left-Maximality of the Second Component

A pair (y, z) is not left-maximal w.r.t. the second component if we can find a pair $(y, z'z)$ with the same number of occurrences and a location list $\mathcal{L}$ whose elements (i, j) are such that the indexes i are unchanged, and the indexes j can be smaller but must respect the constraint $j - i + |y| > 0$.

To intercept the second components that are not left maximal, we consider each tandem tree D_y at a time, along with the list of occurrences of y. We then proceed with an annotation of weights as before, but following a different procedure. If p is an occurrence of y, for each symbol $a \in \Sigma$ we increment the weight of the leaves corresponding to the position $i \in \{p + |y| + 2, \ldots p + d\}$ if and only if $s[i - 1] = a$. For position $p + |y| + 1$ and $p + d + 1$ we assign an ∞ weight. This is because the strings that start immediately after y cannot be further extended without overlapping $|y|$. Moreover, the strings that start at position $p + d + 1$, if extended with a will start at position $p + d$ and would then be counted. However, this would modify the location list, so we need to keep track of this event. By setting the weight of the co-occurrence to infinity we are sure that every component with that occurrence will have a weight different than before the extension. We finally proceed with a bottom-up annotation of the tandem tree. If the newly computed weight at a node β, with $z = \omega(\beta)$, is the same as before, the corresponding tandem $\langle y, z\rangle$ is covered by $\langle y, az\rangle$. We repeat the procedure for each $a \in \Sigma$, and finally traverse the tree eliminating all the nodes that have been marked as covered by one of the tried extensions. Then D_y is traversed to merge paths that have been eventually left with a single child chain. Note that since we are interested just in the detection of not maximal pair, we can limit the extensions to try to one symbol, since this is a sufficient condition to have a tandem that is not maximal.

The annotation of a tandem tree requires $O(n)$ time and must be repeated for $|\Sigma|$ symbols. The number of tandem trees is $O(n)$. Since the size of the alphabet is constant, the overall complexity is again $O(n^2)$.

4.5 Left-Maximality of the First Component

The pair (y, z) is not left-maximal w.r.t. the first component if we can find a string $y'y$ such that $(y'y, z)$ cover the location list of (y, z). Moving the first component to the left alters all the distances in the location list of (y, z) so it might happen that some z falls at a distance bigger than d or that some new occurrence of z appears immediately after the beginning of $y'y$.

Similarly as before, we just need to prove that we can extend the first component of one symbol to the left to prove that the corresponding tandem is not maximal. Let us consider each tandem tree D_y, and the possible extension ay for y, with $a \in \Sigma$. We proceed with a new annotation of D_y according to the occurrence list of ay. If y occurs at position p, we increment the weight of the

position $i \in \{p + |y| + 1 \dots p + d - 1\}$ if and only if $s[p-1] = a$. The positions $p + |y|$ and $p + d$ are assigned an ∞ weight because they will alter the location list. We then proceed with the bottom-up annotation, and if the weight of a node β, with $z = w(\beta)$, has the same weight as before, then the tandem $\langle y, z\rangle$ is covered by $\langle ay, z\rangle$. When all the children have been considered D_y is traversed to merge paths that have been eventually left with a single child chain.

The annotation of a tandem tree requires $O(n)$ time and must be repeated for $|\Sigma|$ symbols. The number of tandem trees is $O(n)$. Since the size of the alphabet is constant, the overall complexity is $O(n^2)$.

4.6 Irredundant Tandem Motif Extraction

From the remaining tandem trees we can delete all the leaves with occurrence count equal to 1 (note that since we do not pose any constraint we can have that two occurrences of y are followed by the same occurrence of z within distance d, thus the value of the leaves is not necessarily 1). For each tandem tree D_y and each node β in it, with $z = w(\beta)$, we report in output the pair (y, z).

From this set $\mathcal{T}$ we can extract the irredundant tandem motifs as follows. Let $\mathbf{L}_\mathcal{T}$ be the collection of the location lists of all the tandem motifs in $\mathcal{T}$. For each $\mathcal{L}_T \in \mathbf{L}_\mathcal{T}$, if $\mathcal{L}_T = \mathcal{L}_{T_1} \bigcup \mathcal{L}_{T_2} \dots \bigcup \mathcal{L}_{T_h}$ up to some offsets, with $T_1, \dots, T_h \in \mathcal{T}$ and $T_i \neq T$ $(i = 1, 2, \dots, h)$, then T is redundant. If, on the other hand, there is no way to express $\mathcal{L}_T$ by other location lists in $\mathbf{L}_\mathcal{T}$, then T is irredundant and we can add it to the output set. This step is afforded in $O(n)$ time for each list (cf., e.g., [11]) by checking whether all occurrences in $\mathcal{L}_T$ falls into the "footprints" of some occurrence of some of the other tandem motifs.

Let M be the number of maximal motifs extracted in $O(n^2)$ with the extensions described above. The time complexity of the last phase is $O(Mn)$. Since the number of maximal motifs is upper bounded by $O(n^2)$, the overall complexity is consequently upper bounded by $O(n^3)$.

5 Concluding Remarks

In this paper we introduced the concepts of maximality and irredundancy for the class of motifs that consists of pairs of co-occurring words, i.e. tandems. We showed that these two properties are not immediately deducible from the german concepts for single words applied to each component. We proved that the number of irredundat not overlapping tandems is linear in the length of the input string, and we gave algorithms to extract such a set.

It is natural to speculate as to whether the present approach can be extended to r-motifs, that are, motifs consisting of r co-occurring solid words, with $r > 2$. To this aim, let $t = \langle m_1, m_2, \dots, m_r\rangle$ be an r-motif, and let d be the maximum allowed distance occurring between each pair (m_i, m_{i+1}) $(1 < i < r-1)$. The notions of occurrence, maximality, irredundancy and exposed occurrence translate with straightforward interpretation for r-motifs.

Along the line of Theorem 2, the following lemma holds.

Lemma 2. Let s be a string of length n on a generic alphabet Σ. Then, the number of irredundant r-motifs $t = \langle m_1, m_2, \ldots, m_r \rangle$ with $r > 2$ non-overlapping solid components in s is $O(d^{2(r-1)}n)$.

Proof. Given a set of positions $I = (i_1, i_2, \ldots, i_r)$ of s, for each i_h $(1 < h < r-1)$ there are at most $d-1$ different subwords that can concur to be a component of some irredundant r-motifs. For each of the $O(d^{r-1})$ resulting combinations of co-occurring subwords, there is at most one subword starting at position i_r that can be the r-th component of an irredundant r-motif with an exposed occurrence at $(i_1, i_2, \ldots, i_r)$. Otherwise, the condition of irredundancy would be contradicted, according to the case $r = 2$. Since the number of sets of positions $I = (i_1, i_2, \ldots, i_r)$ such that $i_{h+1} - i_h \geq d$ $(1 < h < r-1)$ is $O(d^{r-1}n)$ the claim is proved.

□

Finally, being this the first work, to the best of our knowledge, investigating the properties of maximality and irredundancy for tandems as a whole, several questions can be raised from this point and be topic for future research. As an example, the approach presented here can be seen as a first step towards faster and truly efficient algorithms for tandem motif finding, due to the compactness of the proposed motif class. Another point worth attention is studying how the complexity bounds change if we allow the components to have inexact matches in the input string.

References

1. Apostolico, A., Comin, M., Parida, L.: VARUN: Discovering extensible motifs under saturation constraints. IEEE/ACM Trans. Comp. Biol. Bioinf. 7(4), 752–762 (2010)
2. Apostolico, A., Parida, L.: Incremental paradigms of motif discovery. J. of Comp. Biol. 11(1), 15–25 (2004)
3. Apostolico, A., Pizzi, C., Satta, G.: Optimal Discovery of Subword Associations in Strings. In: Suzuki, E., Arikawa, S. (eds.) DS 2004. LNCS (LNAI), vol. 3245, pp. 270–277. Springer, Heidelberg (2004)
4. Apostolico, A., Pizzi, C., Ukkonen, E.: Efficient algorithms for the discovery of gapped factors. Algorithms for Molecular Biology (6:5) (2011)
5. Apostolico, A., Satta, G.: Discovering subword associations in strings in time linear in the output size. J. Discrete Algorithms 7(2), 227–238 (2009)
6. Apostolico, A., Tagliacollo, C.: Incremental discovery of the irredundant motif bases for all suffixes of a string in o(n^{2}logn) time. Theor. Comput. Sci. 408(2-3), 106–115 (2008)
7. Apostolico, A., Tagliacollo, C.: Optimal extraction of irredundant motif bases. Int. J. Found. Comput. Sci. 21(6), 1035–1047 (2010)
8. Carvalho, A.M., Freitas, A.T., Oliveira, A.L., Sagot, M.-F.: An efficient algorithm for the identification of structured motifs in DNA promoter sequences. IEEE/ACM Trans. Comput. Biology Bioinform. 3(2), 126–140 (2006)
9. Fassetti, F., Greco, G., Terracina, G.: Mining loosely structured motifs from biological data. IEEE Trans. Knowl. Data Eng. 20(11), 1472–1489 (2008)

10. Grossi, R., Pietracaprina, A., Pisanti, N., Pucci, G., Upfal, E., Vandin, F.: MADMX: A strategy for maximal dense motif extraction. J. of Comp. Biol. 18(4), 535–545 (2011)
11. Grossi, R., Pisanti, N., Crochemore, M., Sagot, M.-F.: Bases of motifs for generating repeated patterns with wild cards. IEEE/ACM Trans. Comp. Biol. Bioinf. 2(3), 159–177 (2005)
12. Marsan, L., Sagot, M.-F.: Algorithms for extracting structured motifs using a suffix tree with an application to promoter and regulatory site consensus identification. J. of Comp. Biol. 7(3-4) (2000)
13. Marsan, L., Sagot, M.-F.: Extracting structured motifs using a suffix tree - algorithms and application to promoter consensus identification. In: Proceedings of the fourth annual international conference on Computational Molecular Biology (RECOMB), pp. 210–219 (2000)
14. Monteiro, P.T., Mendes, N.D., Teixeira, M.C., others: Yeastract-discoverer: new tools to improve the analysis of transcriptional regulatory associations in *saccharomyces cerevisiae*. Nucleic Acids Research 36(Database-Issue), 132–136 (2008)
15. Mularoni, L., Guigó, R., Albà, M.M.: Mutation patterns of amino acid tandem repeats in the human proteome. Genome Biol. 7(4), R33 (2006)
16. Parida, L., Rigoutsos, I., Floratos, A., Platt, D.E., Gao, Y.: Pattern discovery on character sets and real-valued data: linear bound on irredundant motifs and an efficient polynomial time algorithm. In: Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 297–308 (2000)
17. Pelfrêne, J., Abdeddaïm, S., Alexandre, J.: Extracting approximate patterns. J. Discrete Algorithms 3(2-4), 293–320 (2005)
18. Pisanti, N., Crochemore, M., Grossi, R., Sagot, M.-F.: A Basis of Tiling Motifs for Generating Repeated Patterns and Its Complexity for Higher Quorum. In: Rovan, B., Vojtáš, P. (eds.) MFCS 2003. LNCS, vol. 2747, pp. 622–631. Springer, Heidelberg (2003)
19. Pisanti, N., Crochemore, M., Grossi, R., Sagot, M.-F.: A comparative study of bases for motif inference. In: Iliopoulos, C., Lecroq, T. (eds.) String Algorithmics, pp. 195–226. KCL Publications (2004)
20. Rombo, S.E.: Extracting string motif bases for quorum higher than two. Theor. Comput. Sci (2012)
21. Ukkonen, E.: Maximal and minimal representations of gapped and non-gapped motifs of a string. Theor. Comput. Sci. 410(43), 4341–4349 (2009)