Theoretical Computer Science 525 (2014) 89–102

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

# Irredundant tandem motifs

Laxmi Parida [a], Cinzia Pizzi [b], Simona E. Rombo [c,*]

[a] IBM T.J. Watson Research Center, United States
[b] Department of Information Engineering, University of Padova, Italy
[c] Department of Mathematics and Computer Science, University of Palermo, Italy

ARTICLE INFO

*Keywords:*
Motifs
Tandem
Patterns
Irredundant
Redundant motifs
String algorithm
Suffix tree

ABSTRACT

Eliminating the possible redundancy from a set of candidate motifs occurring in an input string is fundamental in many applications. The existing techniques proposed to extract irredundant motifs are not suitable when the motifs to search for are *structured*, i.e., they are made of two (or several) subwords that co-occur in a text string $s$ of length $n$.

The main effort of this work is studying and characterizing a compact class of *tandem motifs*, that is, pairs of substrings $\langle m_1, m_2 \rangle$ occurring in tandem within a maximum distance of $d$ symbols in $s$, where $d$ is an integer constant given in input. To this aim, we first introduce the concept of *maximality*, related to four specific conditions that hold only for this class of motifs. Then, we eliminate the remaining redundancy by defining the notion of *irredundancy* for tandem motifs.

We prove that the number of non-overlapping irredundant tandem motifs is $O(d^2 n)$ 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.

The notions and bounds provided for tandem motifs are generalized for the case $r \geqslant 2$, if $r$ is the number of subwords composing the motifs. Finally, we also provide an algorithm to extract irredundant tandem motifs.

## 1. Introduction

The problem of generating compact sets of patterns where the possible redundancy, intrinsic of the very nature of repetitive patterns, has been eliminated, was studied w.r.t. different domains, such as biosequence motif discovery (e.g. [1–4]), digital images processing [5–8], graph analysis[1] [9–12]. In all these cases, the *pattern* can be seen as a sub-structure of the input data structure (e.g., a substring, a sub-image or a sub-graph, respectively). None of the above mentioned approaches deals with the problem of extracting irredundant patterns made of two or several sub-structures that often occur together (i.e., *co-occur*) in the input structure.

When we consider text strings as input, extracting pairs (or sets) of co-occurring substrings is an important task in different application contexts, such as for example bioinformatics [13,14] or natural language processing [15]. In the last few years, several approaches have been proposed (e.g., [16–20]) dealing with the most general version of the problem, that is, extracting sets of substrings that occur (also non-exactly) together in a given sequence, within a distance that is fixed in a finite range. Despite their flexibility, such techniques do not address the problem 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

* Corresponding author.
*E-mail address:* simona.rombo@math.unipa.it (S.E. Rombo).

[1] Note that, in case of graph mining, the object is not necessarily connected.

0304-3975/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.tcs.2013.08.012

components in [15,21] a compact index was proposed to compute the number of co-occurrences 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 [22] this bound was improved to the actual size of the output. In [23] 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.

In this work[2] we address the problem of extracting pairs of substrings $\langle m_1, m_2\rangle$ 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 substring pairs.[3]

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. In Section 4 we study the case $r > 2$. Section 5 presents a procedure to extract irredundant tandem motifs. Finally, in Section 6 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 $\Sigma$, 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 string $y$. Given two strings $y_1$ and $y_2$, $y_1 y_2$ indicates the concatenation of $y_1$ and $y_2$, and its length is $|y_1 y_2| = |y_1| + |y_2|$.

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

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

**Definition 3** *(Tandem, tandem occurrence).* Let $d$ be a positive integer (aka *distance*) such that $d \leqslant n$, and $m_1$ and $m_2$ be two substrings of $s$ such that $|m_1| \leqslant d$.

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. $|m_1| \leqslant j - i \leqslant d$;
2. $m_1$ 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 $s$.

Condition 1 also ensures that the occurrences of the two components $m_1$ and $m_2$ do not overlap.

**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 $t''$ ($t' \preccurlyeq t''$) 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 $q \leqslant n$, 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 $\ell_i$ for each $\ell_i \in \mathcal{L}_t$;
2. $p \geqslant q$;
3. there is no pair $\ell \neq \ell_h$, $1 \leqslant h \leqslant p$, such that $t$ occurs at $\ell$ in $s$ (the location list is of maximal size, i.e., it contains all the occurrences of $t$ in $s$).

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 substrings of $m'_1$ and $m'_2$, respectively, and $|\mathcal{L}_t| = |\mathcal{L}_{t'}|$.

[2] A preliminary version of this manuscript has been published in [24].

[3] In [15,22] 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.