Theoretical Computer Science 638 (2016) 76–90

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Sequence similarity measures based on bounded hamming distance ☆

Alberto Apostolico [a,b,1], Concettina Guerra [a,b], Gad M. Landau [c,d], Cinzia Pizzi [e,*]

[a] *College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30318, USA*
[b] *Istituto di Analisi dei Sistemi e Informatica, Consiglio Nazionale delle Ricerche, Roma, Italy*
[c] *Department of Computer Science, University of Haifa, Haifa 31905, Israel*
[d] *Department of Computer Science and Engineering, NYU Polytechnic School of Engineering, New York University, Six MetroTech Center, NY 11201-3840, USA*
[e] *Dipartimento di Ingegneria dell'Informazione, Università degli Studi di Padova, via Gradenigo 6/A, 30131 Padova, Italy*

ARTICLE INFO

Article history:
Received 30 March 2015
Received in revised form 12 January 2016
Accepted 18 January 2016
Available online 21 January 2016

Keywords:
Pattern matching
String comparison
Alignment free distances
Binary string
Longest common substring
Mismatches

ABSTRACT

A growing number of measures of sequence similarity are being based on some underlying notion of relative compressibility. Within this paradigm, similar sequences are expected to share a large number of common substrings, or subsequences, or more complex patterns or *motifs*, and so on. In this paper, measures of sequence similarity are introduced and studied in which patterns in a pair are considered similar if they coincide up to a preset number of mismatches, that is, within a bounded Hamming distance. It is shown here that for some such measures bounds are achievable that are slightly better than $O(n^2)$. Preliminary experiments demonstrate the potential applicability to phylogeny and classification of these similarity measures.

1. Introduction

The comparison and classification of very long textfiles of the kind massively emerging in transmission and digital repositories poses the increasing demand for measures of similarity that are global and fast to implement. In particular, whole genomes analysis resort to global similarity measures based on some implicit notion of mutual compressibility that refer, implicitly or explicitly, to the composition of sequences in terms of various kinds of patterns such as substrings (e.g. [3,8,13,14,17,28,31,21]), subsequences (e.g. [1,2,7,11,22,24]), or motifs (e.g. [16,23,27,30]).

All such methods are based ultimately on quantities that might be loosely referred to as measures of information content [10,12,15], and are sometimes collectively referred to as alignment-free comparisons [29]. Universal measures of the (conditional) information content of finite sequences have been an elusive goal since von Mises' investigation of the notion of randomness [20]. Attempts along this line include Brillouin's adoption of Shannon's redundancy [9], and Kolmogorov's approach to information [18].

☆ A related extended abstract was presented at the 2014 IEEE Data Compression Conference [4].

* Corresponding author.
E-mail addresses: axa@cc.gatech.edu (A. Apostolico), guerra@cc.gatech.edu (C. Guerra), landau@univ.haifa.ac.il (G.M. Landau), cinzia.pizzi@dei.unipd.it (C. Pizzi).

[1] Died on 20th of July 2015.

http://dx.doi.org/10.1016/j.tcs.2016.01.023
0304-3975/© 2016 Elsevier B.V. All rights reserved.

y:	A	C	C	T	G	G	T	A	T	G		
x:			C	T	G	G	A	A	T	C	G	G
$CORR_3$:			1	1	1	1	0	1	1	0		

Fig. 1. The vector $CORR_3$ is obtained by superposing x onto y starting at position 3 in y.

The computational complexity of measures based on strings composition varies, and it increases with the complexion of the patterns taken into account. At the low end of the spectrum, most measures based on the bags of shared *substrings* are typically afforded in linear time. This performance is no longer achievable as soon as some degree of distortion is accepted.

In this paper, we consider extensions of methods based on exhaustive comparisons of substrings, such as the *average common substrings* and the *exhaustive composition vector* method [3,28]. The direction of extension is that of including mismatches, a framework within which algorithms have been proposed, in previous work, to count all the approximate occurrences of fixed length substrings, and to compute statistical expectation for the analysis within a single biological sequence [5,6,25,26].

The rest of the paper is organized as follows. In Section 2 we will introduce sequences similarity measures with mismatches, and in Section 3 we will present algorithms to compute such measures. In Section 4 we will assess the expected length of the maximal and average common match between to sequences. Finally, in Section 5 we will present and discuss the results of some experiments with the proposed measures of similarity with mismatches.

2. Preliminaries

Assume to be given two long sequences x and y. Let n be the length of either one, expressed in terms of the number of characters from a finite alphabet Σ from which both sequences are drawn. Also assigned is an integer k, which is also a parameter but one that can be expected to be in practice much smaller than $\log n$.

Definition 1 *(Cor(i,j)).* $COR(i, j), i = 1, 2, \ldots, n; j \geq i$ is defined as the length of the longest substring beginning at position i in x that can be copied starting from position j in y with exactly k mismatches.

We now consider the following measures of cross correlation.

Definition 2 *(MaxCor).* *MaxCor* is defined as the maximum value attained by *COR* over all values of $i \in (1, 2, \ldots, n)$ in x and $j \in (i, i+1, \ldots, n)$ in y.

Definition 3 *(AvCor).* *AvCor* is defined as the average value attained by *COR* over all values of $i \in (1, 2, \ldots, n)$ of x and $j \in (i, i+1, \ldots, n)$ in y.

Definition 4 *(MaxCor(i)).* $MaxCor(i), i = 1, 2, \ldots, n$ is defined as the maximum value attained by $COR(i, j)$ for each i over all values correspondingly spanned by j.

Definition 5 *(AvCor(i)).* $AvCor(i), i = 1, 2, \ldots, n$: the average value attained by $COR(i, j)$ for each i over all values correspondingly spanned by j.

We observe that the span of j in the definitions above involve an upper triangular matrix. This limitation is chosen only for simplicity of exposition and is removed in practice at no extra cost.

All of the above measures admit of an easy $O(n^2)$ computation. E.g., to compute $MaxCor(i)$, we do the following:

1. First produce the n binary vectors $CORR_i$, $i = 1 \ldots n$, resulting from superposition of x onto y beginning at positions $1, 2, \ldots, n$; to fix ideas, assume that 1 stands for a match and 0 for a mismatch (Fig. 1).
2. Then consider a vector *LENGTH* of size n. At the end of the process $LENGTH(i)$ will store the value of $MaxCor(i)$.
3. In the first step, starting at the first position of $CORR_1$, find the longest string w with k 0's and annotate its length in $LENGTH(1)$ (if no such string exists just abort). Let u be the first run of 1's in w (we assume $|u| > 0$ and leave the case of an empty u to the reader).
4. Observe that by our choice of w it must be

   $$CORR_1(|w| + 1) = CORR_1(LENGTH(1) + 1) = 0$$

   so that $LENGTH(2), LENGTH(3), \ldots, LENGTH(|u| + 1)$ are consecutive unit decrements from $LENGTH(1)$, and we can compute $LENGTH(|u| + 2)$ by moving a *left* pointer on $CORR_1$ past the first 0 falling inside w and advancing a *right* pointer till we hit the next 0 on $CORR_1$.
5. Continue the scan to compute all remaining values of the vector *LENGTH*.
6. Repeat the process for all $CORR_i$ vectors $(i = 2 \ldots n)$, each time suitably updating the value at each position of the vector *LENGTH* to the maximum value found so far.