Approximate string-matching with a single gap for sequence alignment

Tomáš Flouri
Czech Technical University in Prague, Faculty of Information Technology, Dept. of Theoretical Computer Science
Tomas.Flouri@fit.cvut.cz

Kimon Frousios
King's College London
Dept. of Informatics
Strand, London WC2R 2LS, England, United Kingdom
kimon.frousios@kcl.ac.uk

Costas S. Iliopoulos*
King's College London
Dept. of Informatics
Strand, London WC2R 2LS, England, United Kingdom
csi@dcs.kcl.ac.uk

Kunsoo Park
School of Computer Science and Engineering, Seoul National University, Seoul 151-742, South Korea
kpark@theory.snu.ac.kr

Solon P. Pissis
King's College London
Dept. of Informatics
Strand, London WC2R 2LS, England, United Kingdom
solon.pissis@kcl.ac.uk

German Tischler
University of Würzburg
Lehrstuhl für Informatik II
Am Hubland, 97074
Würzburg, Germany
tischler@informatik.uni-wuerzburg.de

ABSTRACT

This paper deals with the approximate string-matching problem with Hamming distance and a single gap for sequence alignment. We consider an extension of the approximate string-matching problem with Hamming distance, by also allowing the existence of a single gap, either in the text, or in the pattern. This problem is strongly and directly motivated by the next-generation re-sequencing procedure. We present a general algorithm that requires $\mathcal{O}(nm)$ time, where n is the length of the text and m is the length of the pattern, but this can be reduced to $\mathcal{O}(m\beta)$ time, if the maximum length β of the gap is given.

1. INTRODUCTION

The problem of finding factors of a text similar to a given pattern has been intensively studied over the last thirty years, and it is a central problem in a wide range of applications, including file comparison, spelling correction, information retrieval, and searching for similarities among biosequences.

This work is directly motivated by the next-generation re-sequencing procedure. The constant advances in sequencing technology are turning whole-genome sequencing into a routine procedure, resulting in massive amounts of DNA and RNA data that need to be processed [3]. Tens of gigabytes of data in the form of short sequences (reads) need to be mapped (aligned) back to reference sequences, a few gigabases long, to infer the read from which the genomic location derived. This is a challenging task because of the high data volume and the size of large genomes. In addition, the performance, in terms of sensitivity, accuracy and speed, deteriorates in the presence of inherent genomic variability and sequencing errors, particularly so for relatively short insertions and deletions (gaps).

A gap can be described as the absence (presence) of a region in one sequence, which is (is not) present in another, as part of the natural diversity between individuals. There are several mechanisms by which gaps can occur in DNA sequences, e.g. inserted or deleted bases caused by slipping of the DNA replication machinery; duplicated or deleted regions caused by the DNA repair machinery; large deletions and duplications caused by uneven recombination between chromosomes; sequencing errors for some platforms.

Concerning the length of the gaps, a very broad range of lengths is possible. In practice, however, the size of reads is too small to confidently detect a large gap directly. In Fig. 1, the distribution of lengths of gaps in exome sequencing is demonstrated[1]. The shape of the distribution of lengths of gaps is consistent with other studies [4]. The presented data reflect a gap occurrence frequency of approximately 5.7×10^{-6} across the exome.

The main observation is the exponential decrease of frequency as length increases, and a preference for multiples of 3. For short reads of length in the order of 100bp the presence of multiple gaps is unlikely given the gap occurrence frequency, and could greatly reduce the mapping confidence of those reads. Hence, applying a traditional dynamic programming approach (see [5] for local alignment or [2] for global alignment), which allows multiple replacements, in-

[*] Prof. Iliopoulos is also affiliated with Curtin University, Digital Ecosystems & Business Intelligence Institute, Centre for Stringology & Applications, GPO Box U1987 Perth WA 6845, Australia.

[1] Data generated by the Exome Sequencing Programme at the NIHR Biomedical Research Centre at Guy's and St Thomas' NHS Foundation Trust in partnership with King's College London.

ACM-BCB '11, August 1-3, Chicago, IL, USA
Copyright 2011 ACM 978-1-4503-0796-3/11/08 ...$10.00.

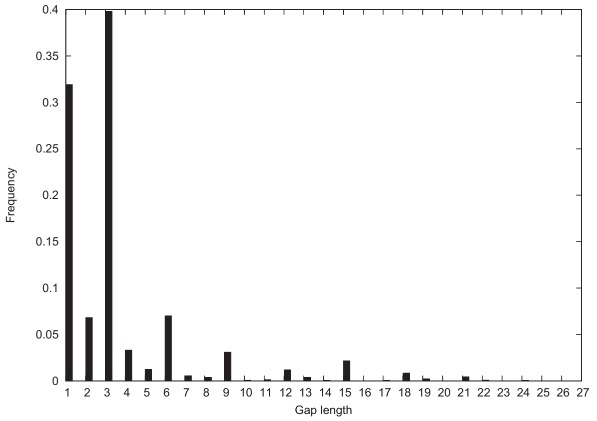

Figure 1: The distribution of lengths of gaps in exome sequencing.

sertions and deletions in different positions of the read or the reference, would affect the confidence of the mapping.

In this paper, motivated by the observations made in the analysis of exome sequencing data, we consider an extension of the approximate string-matching problem with Hamming distance, by also allowing the existence of a single gap, either in the text, or in the pattern. We present a general algorithm that solves this problem in $\mathcal{O}(nm)$ time, where n is the length of the text and m is the length of the pattern, but this can be reduced to $\mathcal{O}(m\beta)$ time, if the maximum length β of the gap is given.

Hence, after aligning a fragment of a short read with a factor of the reference genome as a seed, the proposed algorithm can be directly applied to locally align the remaining part of the read with a relatively short factor of the reference, to allow a number of mismatches and the existence of a single gap, either in the read or in the reference.

2. BASIC DEFINITIONS

A *string* or *sequence* is a succession of zero or more symbols from an alphabet Σ of cardinality s; the string with zero symbols is denoted by ϵ. The set of all strings over the alphabet Σ including ϵ, is denoted by Σ^*. The set Σ^+ is defined as $\Sigma^+ = \Sigma^* \setminus \{\epsilon\}$. A string x of length m is represented by $x[1\,..\,m]$, where $x[i] \in \Sigma$ for $1 \le i \le m$. The length of a string x is denoted by $|x|$. A string w is a factor of x if $x = uwv$ for $u, v \in \Sigma^*$. It is a *prefix* of x if u is empty and a *suffix* of x if v is empty. The Hamming distance δ_H is defined only for strings of the same length. For two strings x and y, $\delta_H(x, y)$ is the number of places in which the two strings differ, i.e. have different characters. A *don't care* symbol $\star$, is a symbol that matches every other symbol. The *don't care* matches every symbol of Σ that is, $\star = a$ for each $a \in \Sigma$. Given an integer γ, $\gamma > 0$, we define a *gap* $g(\gamma)$ as a string of don't care symbols of length γ, i.e. $g(\gamma) \in \{\star\}^+$. A *gap string* y is the concatenation of a sequence of strings over Σ and gaps, i.e. $y \in (\Sigma \cup \{\star\})^*$. Given a gap string $y = y_1 g_1 y_2 g_2 \ldots y_{n-1} g_{n-1} y_n$, such that $y_i \in \Sigma^*$ and $g_i \in \{\star\}^*$, then $c(y) = y_1 y_2 \ldots y_n$. A gap string x *matches* a gap string y with at most k-mismatches, iff $\delta_H(x, y) \le k$ and $|x| = |y|$. A gap string y is called *single gap string*, if it contains only one gap, i.e. $y = y_1 g y_2$, $y_1, y_2 \in \Sigma^*$ and $g \in \{\star\}^+$.

		0	1	2	3	4	5
		ϵ	G	G	G	T	A
0	ϵ	**0**	**0**	**0**	0	0	0
1	A	**0**	**1**	**1**	**1**	1	0
2	G	**0**	**0**	**1**	**1**	**1**	1
3	G	0	**0**	**0**	**1**	**1**	**1**
4	T	0	1	**1**	**1**	**1**	**1**
5	C	0	1	1	**1**	**1**	**2**
6	A	0	1	1	1	**1**	**1**
7	T	0	1	1	1	1	**2**

(a) Matrix G

		0	1	2	3	4	5
		ϵ	G	G	G	T	A
0	ϵ	**0**	**1**	**2**	3	4	5
1	A	**1**	**0**	**0**	**0**	0	0
2	G	**2**	**0**	**0**	**0**	**2**	3
3	G	3	**0**	**0**	**0**	**1**	**2**
4	T	4	0	**0**	**0**	**0**	**1**
5	C	5	0	3	**2**	**1**	**0**
6	A	6	0	4	3	**2**	**0**
7	T	7	0	5	4	0	**0**

(b) Matrix H

Table 1: Matrix G and matrix H for t=AGGTCAT and x=GGGTA

3. PROBLEM DEFINITION

Problem 1. Given a text $t = t[1\,..\,n]$, a pattern $x = x[1\,..\,m]$, $t, x \in \Sigma^*$, $n \ge m$, and integers k, $0 \le k < m$, α and β, $0 \le \alpha \le \beta$, $\beta \le n$, find all the positions of factors of t, such that for each factor, say y

- either there exists a single gap string, say y', with a gap $g(\gamma)$, $\gamma > 0$, such that $y = c(y')$, $\delta_H(x, y') \le k$, and $\alpha \le \gamma \le \beta$
- or there exists a single gap string, say x', with a gap $g(\gamma)$, $\gamma > 0$, such that $x = c(x')$, $\delta_H(x', y) \le k$, and $\alpha \le \gamma \le \beta$
- or $\delta_H(x, y) \le k$ and $\alpha = 0$

4. THE ALGORITHM

The initial focus is on computing a matrix $G[n+1, m+1]$, where $G[i, j]$ contains the minimum number of mismatches of the factor $t[1\,..\,i]$ of the text, and the factor $x[1\,..\,j]$ of the pattern, with at most one gap, either in the text, or in the pattern.

Example. Let the text t=AGGTCAT and the pattern x=GGGTA. Table 1a shows matrix G.

In order to compute the location of the gap either in the text or in the pattern, we also need to compute a matrix $H[n+1, m+1]$ such that,

$$H[i,j] = \begin{cases} \text{if } G[i,j] = G[i,i] \text{ and } i \le j, & H[i,j] = j - i \\ \text{if } G[i,j] = G[j,j] \text{ and } i > j, & H[i,j] = i - j \\ \text{otherwise,} & H[i,j] = 0 \end{cases} \tag{1}$$

Example. Let the text t=AGGTCAT and the pattern x=GGGTA. Table 1b shows matrix H.

The Gap-Mismatches algorithm for computing matrix G and matrix H is outlined in Figure 2.

Theorem 1. *Given the text $t = t[1\,..\,n]$ and the pattern $x = x[1\,..\,m]$, the* Gap-Mismatches *algorithm can compute matrix G correctly in $\mathcal{O}(nm)$ units of time.*

Proof. Without loss of generality, assume that we want to compute $G[i, j]$, which contains the minimum number of mismatches of $t[1\,..\,i]$ and $x[1\,..\,j]$, with at most one gap.

Let $i < j$. The minimum number of mismatches can be computed by the following,

```
Gap-Mismatches
▷Input: $t$, $n$, $x$, $m$
▷Output: $G$, $H$
 1 begin
 2   ▷ Initialisation
 3   for $i \leftarrow 0$ until $n$ do $G[i,0] \leftarrow 0$; $H[i,0] \leftarrow i$
 5   for $j \leftarrow 0$ until $m$ do $G[0,j] \leftarrow 0$; $H[0,j] \leftarrow j$
 7   ▷ Matrix $G$ and Matrix $H$ computation
 8   for $i \leftarrow 1$ until $n$ do
 9     for $j \leftarrow 1$ until $m$ do
10       if $i < j$ then
11         $u \leftarrow G[i-1,j-1] + \delta_H(t[i],p[j])$
12         $v \leftarrow G[i,i]$
13         $G[i,j] \leftarrow min(u,v)$
14         if $v < u$ then $H[i,j] \leftarrow j-i$
15         else $H[i,j] \leftarrow 0$
16       if $i > j$ then
17         $u \leftarrow G[i-1,j-1] + \delta_H(t[i],p[j])$
18         $v \leftarrow G[j,j]$
19         $G[i,j] \leftarrow min(u,v)$
20         if $v < u$ then $H[i,j] \leftarrow i-j$
21         else $H[i,j] \leftarrow 0$
22       if $i = j$ then
23         $G[i,j] \leftarrow G[i-1,j-1] + \delta_H(t[i],p[j])$
24         $H[i,j] \leftarrow 0$
25 end
```

Figure 2: The Gap-Mismatches algorithm for computing matrix G and matrix H.

$$G[i,j] = \min(\delta_H(t[1\,..\,i], x[j-i+1\,..\,j]), \delta_H(t[1\,..\,i], x[1\,..\,i])) \quad (2)$$

In this case, we take the minimum between the Hamming distance of $t[1\,..\,i]$ and the suffix of x, $x = x[j-i+1\,..\,j]$, and the Hamming distance of $t[1\,..\,i]$ and the prefix of x, $x[1\,..\,i]$, while the suffix of x, $x[i+1\,..\,j]$ is considered to be a gap in the text.

Let $i > j$. The minimum number of mismatches can be computed by the following,

$$G[i,j] = \min(\delta_H(t[i-j+1\,..\,i], x[1\,..\,j]), \delta_H(t[1\,..\,j], x[1\,..\,j])) \quad (3)$$

Similarly, in this case, we take the minimum between the Hamming distance of the suffix of t, $t[i-j+1\,..\,i]$ and $x = x[1\,..\,j]$, and the Hamming distance of the prefix of t, $t[1\,..\,j]$, and $x[1\,..\,j]$, while the suffix of t, $t[j+1\,..\,i]$ is considered to be a gap in the pattern.

Trivially, in the case that $i = j$,

$$G[i,j] = \delta_H(t[1\,..\,i], x[1\,..\,j]) \quad (4)$$

The Equations 2, 3, 4 are computed by the Gap-Mismatches algorithm in lines 13, 19, and 23, respectively. Hence, this algorithm can compute matrix G in $\mathcal{O}(nm)$ units of time. □

As of Theorem 1, starting the trace-back from cell $H[s_i, s_j]$, for some $0 \leq s_i \leq n$, $0 \leq s_j \leq m$, gives a solution to Problem 1, iff $G[s_i, s_j] \leq k$ and the following hold,

- if $s_i < m$, then $s_j = m$, $\alpha \leq m - s_i \leq \beta$; there exists a gap of length $m - s_i$ in the text
- if $s_i > m$ then $s_j = m$, $\alpha \leq s_i - m \leq \beta$; there exists a gap of length $s_i - m$ in the pattern
- if $s_j < m$ then $s_i = n$, $\alpha \leq n - s_j \leq \beta$; there exists a gap of length $n - s_j$ in the pattern
- if $s_i = m$, then $\alpha = 0$; there is no gap

Finally, we can easily compute the position of the gap by using matrix H. However, since the threshold β is given, a pruned version of matrix G and matrix H can be computed in $\mathcal{O}(m\beta)$ time and space, similarly as shown in [1].

Lemma 2. *There exist at most $2\beta + 1$ cells of matrix G, that give a solution to Problem 1.*

Proof. Let the cell $G[s_i, s_j]$, contain the minimum number of mismatches of the text $t[1\,..\,n]$, and the pattern $x[1\,..\,m]$, with at most one gap of maximum length β, either in the text, or in the pattern. Since the length of the gap is either $m - s_i$, if $s_i < m$ and $s_j = m$, or $s_i - m$, if $s_i > m$ and $s_j = m$, or $n - s_j$, if $s_j < m$ and $s_i = n$, then it holds that

- if $\beta + m \leq n$, then $m - \beta \leq s_i \leq \beta + m$ and $s_j = m$ There exist exactly $2\beta + 1$ such cells.
- if $\beta + m > n$, then
  - if $m - \beta \leq s_i \leq n$, then $s_j = m$
  - if $n - \beta \leq s_j < m$, then $s_i = n$

  There exist exactly $2\beta + 1$ such cells.

Since we also have to check whether $G[s_i, s_j] \leq k$, the Lemma holds. □

In addition, $G[i,j]$ depends only on either $G[i-1,j-1]$ and $G[i,i]$, if $i < j$, or $G[i-1,j-1]$ and $G[j,j]$, if $i > j$, or $G[i-1,j-1]$, if $i = j$ (see lines 13, 19, and 23 in Fig. 2, respectively). Hence, we only need to compute a diagonal stripe of width $2\beta + 1$ in matrix G and matrix H, as shown in Tables 1a and 1b in bold, respectively, for the case of t=AGGTCAT, x=GGGTA and $\beta = 2$. As a result, the Gap-Mismatches algorithm can easily be modified to compute the pruned version of matrix G and matrix H in $\mathcal{O}(m\beta)$ time and space.

5. REFERENCES

[1] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology.* Cambridge University Press, New York, NY, USA, 1997.

[2] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443 – 453, 1970.

[3] S. C. Schuster. Next-generation sequencing transforms today's biology. *Nat. Methods*, 5:16–18, 2008.

[4] M. A. Simpson, M. D. Irving, E. Asilmaz, M. J. Gray, D. Dafou, F. V. Elmslie, S. Mansour, S. E. Holder, C. E. Brain, B. K. Burton, K. H. Kim, R. M. Pauli, S. Aftimos, H. Stewart, C. A. Kim, M. Holder-Espinasse, S. P. Robertson, W. M. Drake, and R. C. Trembath. Mutations in NOTCH2 cause Hajdu-Cheney syndrome, a disorder of severe and progressive bone loss. *Nat Genet*, 2011.

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