On minimisers

CACAO

March 26, 2019

1 Introduction

A basic string processing problem, with applications in many areas, consists in finding approximate occurrences of a query string, or *pattern* P, in a target text T (or collection of texts) typically of much greater size. For instance in computational biology, one is interested in mapping DNA sequence fragments to homologe regions in genomes of other species.

When the text is fixed, an *index* $I(T)$ is built in order to allow for finding pattern occurrences without having to scan through the whole text. Full indexes such as the suffix tree or the suffix array represent all substrings of T, but this may come at an unaffordable memory cost. Simpler, partial indexes have been proposed to represent only certain subsequences of a fixed length k, called *kmers*, along with pointers to their occurrences in T. When searching for approximate matches of P in T, the index is first queried for exact matches in the text of a kmer substring of P, called *seed*. Seed matches are then extend from both flanks to confirm or dismiss actual occurrences of the pattern as a whole. This strategy is known as *seed and extend*.

There is a tradeoff between the indexed kmers length and the size of the index, as the number of distinct kmers grows exponentially with k. On the other hand, there is also a compromise between k and the sensitivity and specificity of the index. A small k implies in less indexed kmers, but each kmer will have more occurrences in the text, which may render the index too sensitive but not very specific due to spurious small seed matches, with obvious negative repercussions for the search time efficiency. Conversely, larger values of k may result in more indexed kmers, but the index may then become too specific, leading to significant approximate pattern occurrences being overlooked because of excessively stringent exact seed matching requirements. It is clear that choosing a value for k that properly balances index size with search time and accuracy is an important problem.

Minimisers

Consider a text $T = t_0 \cdots t_{n-1}$ of length n, and let $T[i : j] = t_i \cdots t_{j-1}$ denote the substring of length $j - i$ starting at position i. One such substring of length $w+k-1$ (i.e. $j = i+w+k-1$) corresponds to a window of w successive (overlapping) kmers. Let us call this a (w, k)-*window*, or simply *window*, when the values of w and k are clear by the context. By considering any given ordering of the kmers, e.g. lexicographic, we call the smallest kmer of this window its (w, k)-*minimiser* or, by the same token, simply *minimiser*. Figure 1 ilustrates this concept.

Schleimer et al [**?**] and, independently Roberts et al [**?**] have proposed the use of minimisers for reducing the space of text indexes. The basic idea is as follows. First we slide a (w, k)-window over T and, for each window position, store its minimiser along with its occurrences in a table. Then, when searching for occurrences of a pattern P, we similarly slide a $(w-k)$-window over P and use each minimiser found along the way as a seed for the pattern alignment. Indeed if P and T share a common substring of length at least $w + k - 1$, then its minimiser must be

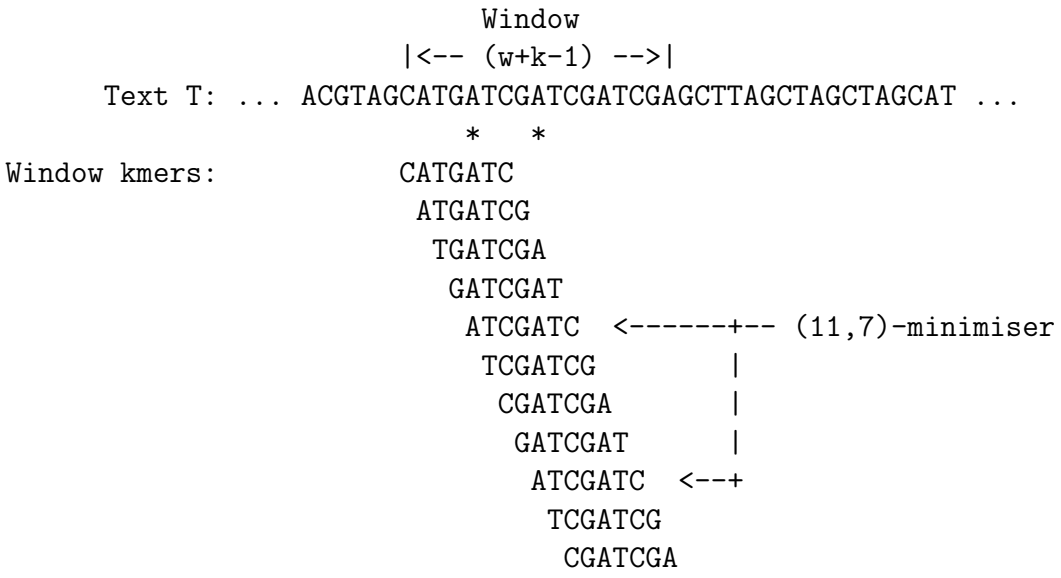

Figure 1: Example of a lexicographic minimiser for $w = 11, k = 7$. Positions where the minimiser occur in the text are indicated by $*$. Note that the minimiser may occur multiple times in the window, possibly with overlaps.

in the table and a seed match will be found. Conversely, any aproximate occurrence of P in T, with an exact alighment of a subsequence of length at least $w + k - 1$ will be pointed out by the index. Thus values w and k jointly control the sensitivity to specificity ratio.

In the seminal works [**?**, **?**] it has also been shown that, under the hypothesis of independence and equiprobalility of the choice of any kmer or a $(w+1, k)$-window as its minimiser, on average a new minimiser comes out at each $(w + 1)/2$ successive window displacements. Therefore w also controls the "sampling rate" of the indexed kmers and consequently the size of the index. It follows that the efficiency and accuracy of the index is heavily dependent on this parametrisation.

However, in some situations, the optimal choice of w and k may prove elusive. In computational biology, for instance, many times the fragments to be mapped (the patterns) vary considerably in terms of length, composition and error rate. Likewise, different regions of the genome may have varying levels of complexity, like regions with long homopolimers or regions enriched with small repeats. Depending on the combination of the characteristics of the patterns and/or regions of the text, it may be insteresting to use different values of w and k. In practice, this often amounts to running a mapping tool multiple times with different parametrisations.