Approximate Hashing for Bioinformatics

Guy Arbitman* Shmuel T. Klein* Pierre Peterlongo** Dana Shapira‡

*Bar Ilan University
Dept. of Computer Science
Ramat-Gan 52900, Israel
guy20495@gmail.com
tomi@cs.biu.ac.il

**Inria, Univ Rennes,
CNRS, IRISA
F-35000 Rennes, France
pierre.peterlongo@inria.fr

‡Ariel University
Dept. of Computer Science
Ariel 40700, Israel
shapird@g.ariel.ac.il

A particular form of lossless data compression is known as *deduplication*, which is often applied in a scenario in which a large data repository is given and we wish to store a new, updated, version of it, in which the changes account only for a tiny fraction of the accumulated information. The idea is then to find duplicated parts and store only one copy P of them; the second and subsequent occurrences of these parts can then be replaced by pointers to P.

One of the approaches to solve the problem is based on classical hashing with the property that when changing even a single bit of the file, the resulting hash value should be completely different, which may not be appropriate in our case. This lead to the design of what could be called an *Approximate Hash* function. The current work is an extension, which applies similar techniques to string processing problems arising in Bioinformatics.

We concentrate on the following two problems. The first problem is that of *clustering* a large collection of DNA strings into sub-collections forming clusters, in the sense that strings assigned to the same cluster may be considered as similar for practical biological purposes, whereas strings of different clusters are different enough to be judged not originating from the same source. The second problem is that of locating a single string within a large collection on the basis of one of its fragments, or rather, one of its fragments that has undergone some limited number of mutations. We show how our notion of an approximate hash may be adapted to these and similar problems.

The idea is to produce a signature encapsulating the main features of the strings in a small number of bits by devising an *occurrence map* of the various substrings of length k, called k-mers, for $k \geq 1$. The ith bit of the signature, corresponding to the i-th ordered k-mer, will be set to 1 if and only if the number of occurrences of this i-th k-mer within the given string is larger than some predetermined threshold t_k, depending on their average number of occurrences. This definition tries to catch underlying similarities, but to remain flexible enough to allow some fluctuations.

The new techniques have been applied on artificial and real-life DNA strings, and compared to state of the art clustering methods. The outcome of the new procedure is very similar to the clustering and search results obtained by accurate tools, but in much less time and with less required memory.