A new method of finding similarity regions in DNA sequences

Laurent Noé — LORIA/UHP Nancy, France
Gregory Kucherov — LORIA/INRIA Nancy, France
Corresponding email : laurent.noe@loria.fr

Similarities

• Identifying similarity regions inside a DNA sequence (repeats), or between two sequences (local alignment), is a fundamental problem in bioinformatics.

• Detecting similarities is a necessary step in functional prediction, phylogenetic analysis, and many other biological studies.

Heuristic Search

• Searching for approximate repeats in whole genomes using exhaustive techniques takes a prohibitive time. Many algorithms find first small exact repeats, called *seeds* (by using suffix tree or hash function), and then try to extend those repeats into approximate ones.

• Our method chains together multiple *seeds* in order to form rapidly large similarity regions, instead of extending individual *seeds*. The choice of search parameters is based on a statistical analysis.

A more sensitive approach

• Grouping *seeds* has been used in a very restricted form by late versions of BLAST[2] ('two-hit' method).

• Our chaining algorithm allows for a smaller *seed* size and therefore for a more sensitive search, without considerable drop in time efficiency. To illustrate the gain in sensitivity, in two sequences of length 100 with at least 66% of similarity, one finds more frequently 3 *seeds* of size 7 than one single *seed* of size 11 (see Figure below).

Method:

Parameters used in the chaining algorithm are estimated according to probability distributions, assuming a Bernoulli model of DNA sequence.

• Maximal distance ρ between *seeds* inside one repeat copy is computed according to *waiting time distribution*[1].

• *Indels* are accounted for by computing a statistical bound δ of *random walk distribution*, which simulates the variation of distance between corresponding *seeds* inside a repeat copy[1].

Results:

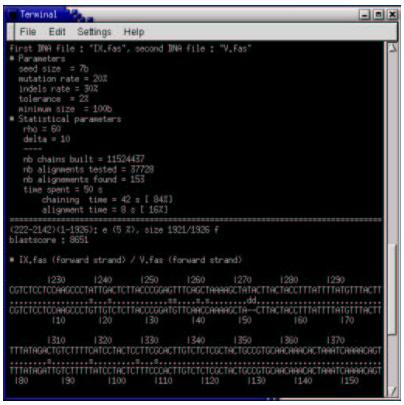
Output is given by start and end positions of each copy, that can be located on either strand. An equivalent BLAST score is also given. Copy alignments can be computed very efficiently due to *seeds* already found.

Experiments:

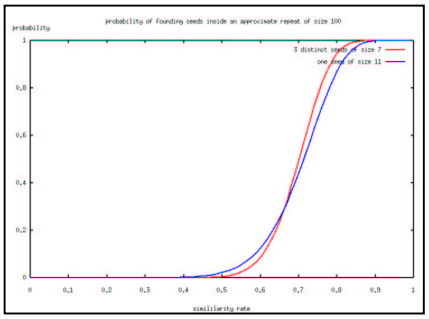
Experiments have been carried out on chromosomes *V* and *IX* of *Saccharomyces cerevisiae*.

Comparative tests have been done with **REPuter** and **BLAST**. They have shown that our program is more sensitive than **BLAST** in finding repetitions with a good similarity rate (70%). **REPuter** tends to keep apart fragments of one repetition whereas our program assembles fragments provided that *indels* are smaller than δ.

Typical Output



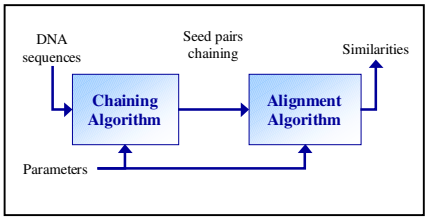
Multiple seeds *vs* single seed



Running Time

Seed size	ρ	δ	*# of alignments*	*# of reps found*	*time*
9	85	10	212	60	3 s
8	64	9	701	64	8 s
7	47	7	748	68	27 s
6	35	6	2388	72	100 s
4	25	5	7863	72	425 s

Algorithm Structure



Seeds

• A *seed* consists of two identical words, one from each sequence. *Seeds* are obtained from a *linked list* computed using a hash function. The list contains all positions of each word in the sequence.

• Two *seeds* are chained together if the distance d between them is bounded by ρ (computed according to statistical criteria, see above) and variation of this distance Δd between two sequences is bounded by δ. These two criteria allow us to chain *seeds* occurring within the same similarity region.

Chaining Algorithm

Chaining algorithm groups together *seeds* involved in the same similarity region. A major problem is to efficiently *retrieve* previously found *seeds* which should be chained with the current *seed*.

This is done by maintaining a **diagonal table** of *seeds*. It stores the last found *seed* indexed by its *diagonal* (difference between the copy positions of the *seed*)

When a new *seed* is found, the chaining algorithm has to look around its *diagonal* to check if another seed can be chained to the new one according to statistical criteria. If no new seed occurred within distance ρ to the right of the last seed of the chain, then the chain is completed and removed from the memory.

The chaining algorithm is linear in the number of *seeds* found, and allows to treat sequences of millions of bp on a regular PC.

References:

[1] G.Benson, *Tandem repeats finder: a program to analyze DNA sequences*, Nucleic Acids Research, 1999, vol 27, 2, pp 573-580.
[2] S.Altschul, W.Gish, W.Miller, E.Myers, D.Lipman, *Basic Local Alignment Search Tool*, Journal of Molecular Biology, 1990, vol 215, pp 403-410.