BLAT

(BLAST-like alignment tool)

Peer review

Ómar Páll Axelsson

opa2@hi.is

BLAT is a high-speed sequence alignment algorithm that can be used to align DNA sequences as well as protein and translated nucleotide (mRNA or DNA) sequences. BLAT quickly finds similarity in DNA and protein, but it needs an exact or nearly-exact match to find a hit. It was originally designed to align three million ESTs and align 13 million mouse whole-genome random reads against the human genome in less than two weeks' time on a moderate-sized Linux cluster. This is the main purpose of developing the BLAT algorithm, so this would also be the main research question: "Can a large number of mouse whole-genome random reads be aligned against the human genome in a much shorter amount of time than with the state of the art algorithms?"

The data that BLAT works with is an index derived from the assembly of an entire genome. By default, the index consists of all non-overlapping 11-mers except for those heavily involved in repeats, and it uses less than a gigabyte of RAM. BLAT is supposedly more accurate and 500 times faster than popular existing tools of its time for mRNA/DNA alignments and 50 times faster for protein alignments at sensitivity settings typically used when comparing vertebrate sequences.

Several other alignment tools are mentioned as well as their respective study is referenced in the introduction and their specific purpose is described briefly. Since BLAT is an abbreviation for "BLAST-like alignment tool", it would be natural to describe BLAST in a fair amount of detail and compare the two. This is done in the second half of the introduction where the main differences between the two are described such as that BLAST builds an index of the query sequence and then scans linearly through the database while for BLAT it's the opposite, BLAT first builds an index of the database and then scans linearly through the query sequence. The description of the differences between the BLAT and BLAST is fairy detailed and gives the reader a decent understanding of why BLAT would be a preferred method in certain cases. BLAT is clearly based on BLAST, so it is not entirely original, but the advantages BLAT is claimed to have makes the study interesting and worth reading.

BLAT was tested against Sim4 which was supposedly good at cDNA alignment. Both BLAT and Sim4 were given the same data of 713 mRNAs corresponding to genes on chromosome 22 and did BLAT's accuracy was slightly better, 99.99% versus Sim4's 99.66% but the remarkable thing was that BLAT took only 26 seconds while Sim4 took almost five hours. Blat was 671 times faster so even faster than the 500 times faster it was claimed to be.

Blat was also run against WU-TBLASTX on a modest-sized data set of 1000 mouse reads against a human chromosome 22 and a nice table is shown displaying the time it took BLAT and WU-TBLASTX on different settings where BLAT was about 30-73 times faster. However, WU-TBLASTX works best on large sequence databases which makes the comparison a bit unfair. The sensitivity of WU-TBLASTX and BLAT was also applied to 13 million mouse shotgun reads and human chromosome 22 where BLAT looked a bit better. This was the dataset that the algorithm was originally built to use, so it is entirely logical to test on that dataset.

In the methods chapter it is explained that the problem must be broken down into two parts, a "search stage" where the regions that are most likely to be homologous are detected. The second stage is the "alignment stage" where the regions are examined in more detail and determined if they are homologous according to a certain criteria.

A few different searching methods are explained in detail and each method has a corresponding table showcasing the percentage of homologies detected for K-mer sizes and percentage identities between the homologous sequences.

The implementation of the match criteria is explained and the sort algorithm that is used, "mSort", is mentioned but not explained.

For the alignment stage both the algorithm for nucleotide alignments and protein alignments are explained. For the nucleotide alignments the algorithm generates a hit list between the query and the homologous region of the database, it looks for relatively small, perfect hits. It recursively fills in gaps that may be in the alignment, using a smaller k each time. For the protein alignment a score function is used to maximally scoring ungapped alignments from the hits of the search stage. A dynamic program maximizes the score from a graph built from the ungapped alignments.

The methods are quite well explained so a reader could in fact reimplement these algorithms for them self.