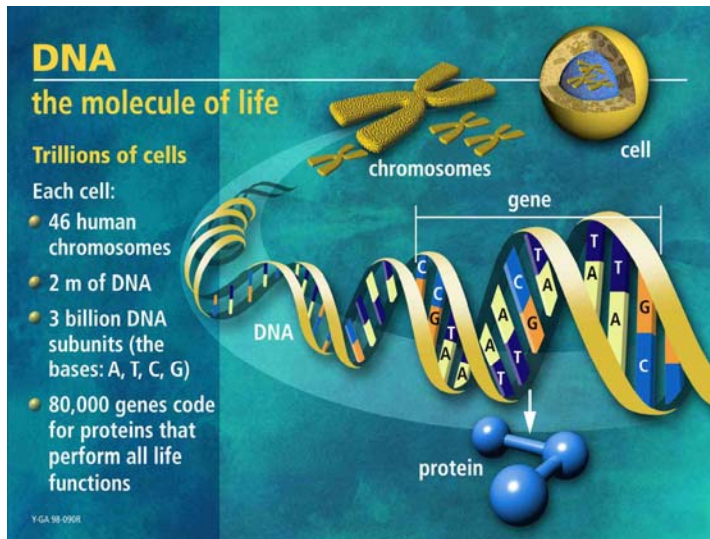


Bio-sequence Similarity Search

By Chen Chang

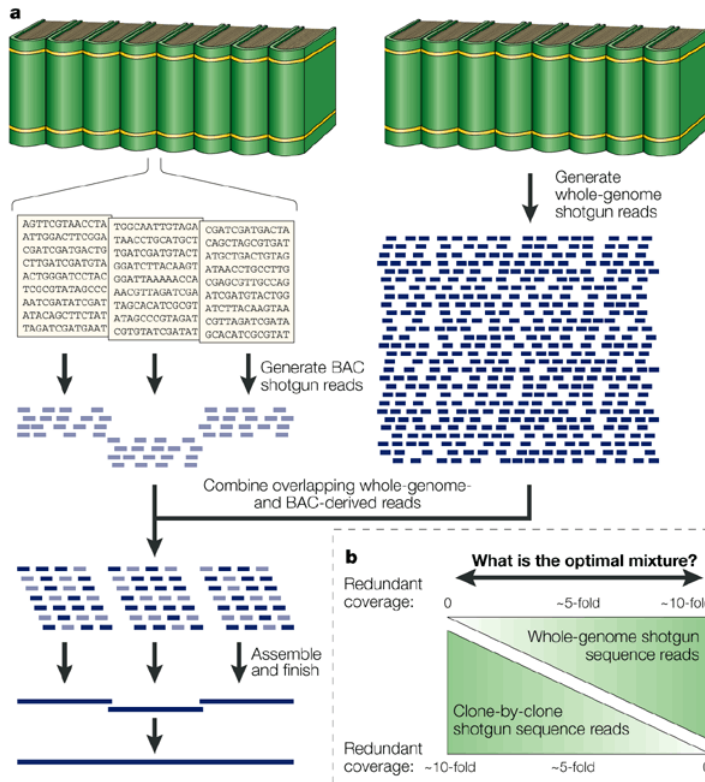
Introduction:



Genomics is the study of how genes and genetic information are organized within the genome, and how this organization determines their function. This science was given an impetus by the Human Genome Project, which stimulated the development of efficient and cheap sequencing techniques. A number of microbial genomes have already been sequenced, followed closely by simple eukaryotic genomes like yeast and the

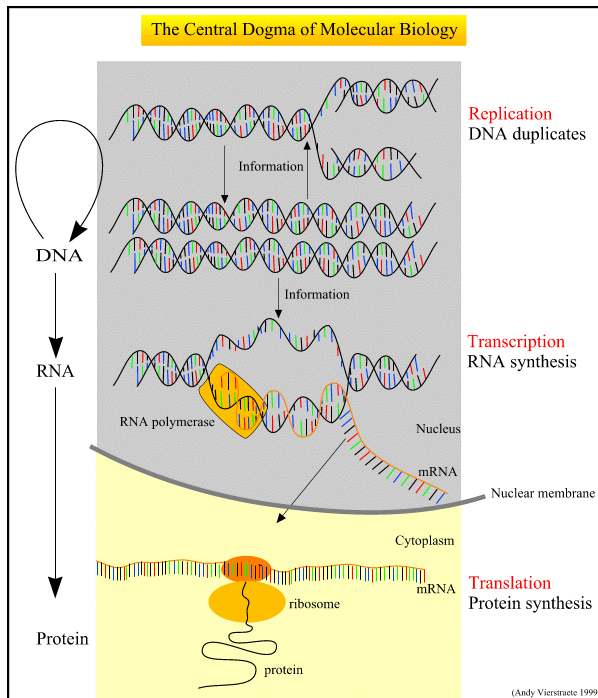
nematode *Caenorhabditis elegans*.

Shotgun genome sequencing is a sequencing strategy for which parts of DNA are randomly sequenced. The sequences obtained have a considerable overlap and by using appropriate computer software it is possible to compare sequences and align them to build larger units of genetic information. This sequencing strategy can be automated and leads to rapid sequencing information, but it is less precise than a systematic sequencing approach.



DNA contains the complete genetic information that defines the structure and function of an organism. Proteins are formed using the genetic code of the DNA. Three different processes are responsible for the inheritance of genetic information and for its conversion from one form to

another:



1. **Replication:** a double stranded nucleic acid is duplicated to give identical copies. This process perpetuates the genetic information.
2. **Transcription:** a DNA segment that constitutes a gene is read and transcribed into a single stranded sequence of RNA. The RNA moves from the nucleus into the cytoplasm.
3. **Translation:** the RNA sequence is translated into a sequence of amino acids as the protein is formed. During translation, the ribosome reads three bases (a codon) at a time from the RNA and translates them into one amino acid

In eucariotic cells, the second step (transcription) is necessary because the genetic material in the nucleus is physically separated from the site of protein synthesis in the cytoplasm in the cell. Therefore, it is not possible to translate DNA directly into protein, but an intermediary must be made to carry the information from one compartment to another.

However, knowing only the sequence of the DNA is still long way from understanding the functionality of individual gene or the protein produced by it. Commonly researchers compare the sequence of an unknown gene/protein to that of a known database of genes/proteins, in order to infer functionality, homology, etc. This process is commonly known as similarity search.

An alignment is an arrangement of two sequences opposite one another, which shows where they are different and where they are similar. The goal is to find the optimal alignment: the most similarity and least differences. Alignment can be measured on both quantity and quality, where quantity refers to the degree of the similarity between the sequences, and quality refers to the specific regions where similarity occurs. There are many methods for alignment, including pattern matching, Hidden Markov Models (HMM), and Smith-Waterman algorithm.

Problem:

Sequence similarity searches are used to search a sequence database for proteins with are homologous to a query protein. The current protein database contains of the order of 100 million residues. For searching with many different sequences, time rapidly becomes an important issue. For this reason, there have been many attempts to produce faster algorithms than straight dynamic programming. The goal of these methods is to search as small a fraction as possible, while still looking at all the high scoring alignments. In cases

where sequences are very similar, there are a number of methods based on extending computer science exact match. However, to find distant matches, these exact methods become intractable, and heuristic approaches has to be used, that sacrifice some sensitivity, in that there are cases, where they can miss the best scoring alignment. A number of heuristic techniques are available. Most popular programs are: [BLAST](#) and [FastA](#).

BLAST (Basic Local Alignment Search Tool) is a set of similarity search programs designed to explore all of the available (DNA and protein) sequence databases regardless of whether the query is protein or DNA. The BLAST programs have been designed for speed, with a minimal sacrifice of sensitivity to distant sequence relationships. BLAST uses the concept of a "segment pair" which is a pair of sub-sequences of the same length that form an ungapped alignment. The algorithm first looks for short words that are present in both sequences and then extend these at either end to find the longest segments present in both. The statistical significance of these High-scoring Segment Pairs is evaluated to determine whether the matches are random or not. Thus, the scores assigned in a BLAST search have a well-defined statistical interpretation, making real matches easier to distinguish from random background.

To speedup the performance of the BLAST algorithm, an MPI implementation of BLAST was developed at Los Alamos National Lab. [mpiBLAST](#) segments the BLAST database and distributes it across cluster nodes, permitting BLAST queries to be processed on many nodes simultaneously. mpiBLAST is based on MPI, typically running on large clusters and micro-processor based supercomputers. When each of the database segments fits in the physical memory of the computing node, the speedup achievable through mpiBLAST can be super-linear. Scalability is limited by the implicit barrier before assembly of the final results. Because database segmentation does not create heavy communication demands, BLAST users can take advantage of low-cost and efficient Linux cluster architectures such as the bladed [Beowulf cluster](#).

Other approaches to speed up the execution of various BLAST algorithms involve the usage of alternative computing hardware, such as ASIC or FPGA. Companies, like [Paracel](#) and [TimeLogic](#), have introduced dedicated hardware accelerator that significantly speedup the BLAST algorithm execution time by a factor of 100s to 1000s over typical cluster computer solutions. However, the excessive cost of such hardware systems and the limited programmability of the software environment have largely limited the practical usage of these systems to a few isolated cases.