Data Mining Technologies for Bioinformatics Sequences

Deepak Garg
Computer Science and Engineering Department
Thapar Institute of Engineering & Tecnology, Patiala

Abstract

Main tool used for sequence alignment search is BLAST. It uses heuristic approach of algorithm for sequence alignment. Underlined algorithm is disscussed with its parameter. Which parameters affect the effectiveness of search algorithm and how these parameters can be improved. Types of sequence databases that are available, the need for different types of sequence databases and importance of algorithms for sequences database search is given. Issues like is there any need of the algorithms for search in sequence databases is explained. Algorithmic issues like space and time reduction using dynamic programming is also explained. How further improvements could be made in this field. Non-algorithmic issues like statistical significance are explained.

1. Introduction

Today the most powerful method for inferring the biological function of a gene is by sequence similarity searching on protein & DNA sequence databases. Thus large-scale sequence comparison, usually organized as database search, is a very powerful tool for biological inference in modern molecular biology. And that tool is almost universally used by molecular biologists. It is now standard practice whenever a new gene is cloned and sequenced, to translate its DNA sequence into an amino acid sequence & then search for similarity between it & members of the protein database. No one today would ever think of publishing the sequence of newly cloned gene without doing such database searches. The following quote will reflect the total impact of sequence databases in field of bio-informatics.

"The new paradigm now emerging is that all the genes will be known (in the sense of being resident in databases available electronically), & that the starting point of biological investigation will be theoretical. An individual scientist will begin with a theoretical conjecture, only then turning to experiment to follow or test that hypothesis".

So search in sequence database is the starting point for any biological investigation like sequence comparison. Given the effectiveness of sequence comparison in molecular biology, it is natural to stockpile and systematically organize the biosequences to be compared; this has naturally led to growth of sequence databases. The secondary purpose of sequence databases is that massive amount of sequence data requires new tools-computers & programs, databases to generate, proof, store and access the data.

As "Necessity Is the Mother of Invention" so there is need of automated data collection, storage tools for managing huge amount of biological information, that is available every new second. Each new second new sequence is reported and they need to get stored so that they can be used in future.

2. Database Types

Broadly, it can be divided into two categories: primary & secondary databases. Primary database contain original biological data such as DNA sequence, or protein structure information from crystallography. Secondary databases attempt to add value to primary databases and make them more useful for certain specialist applications, e.g. PROSITE, the database of common structural or functional motifs found in proteins.

Primary Database

DNA Database: The major DNA archives hold sequences whose length is total more than 500,000,000 base pairs, obtained from parts of 300,00 different genes in many diverse organisms. These numbers have been growing exponentially, and the standard ballpark figure is that number of base pairs held in DNA archives increases by about 75% each year. This speed of growth has important consequences for database strategies. Three groups now collaborate worldwide to produce computerized DNA sequence databases. The EMBL, Genbank, NCBI. These groups exchange information daily and have agreed common standards for DNA sequence database entries.

Genome Database: A second major primary source of primary data is the various genome projects. A number of genome projects are completed included a representative eukaryote, an archeon and three prokaryotes. Much of the information about these projects is now available on EMBL. A large number of genome projects are now underway.

Protein Sequence Database: Two main protein sequence database have grown up to mirror the EMBL and Genbank databases. The SWISS-PORT database consists of properly checked and annotated translations of sequences in the EMBL database. Over years the growth of these databases has been increased exponentially.

Secondary Databases

PROSITE: It is the best-known motif database. It is a dictionary of sites and patterns in proteins that is linked to the protein sequences databases SWISS-PORT. The goal of PRO-SITE is to identify and represent biologically significant patterns in protein families that allow new protein sequences to be reliably assigned to proper family. Searching a sequence against the PROSITE database is very straightforward; simply paste the appropriate sequence into the form. Sites allowing PROSITE searches are available throughout the world. It should always to be remembered that PROSITE is rather blunt sequence analysis instrument. However, PROSITE does contain an extensive collection of patterns and one of the tools that should always be used in function prediction.

3. Underlying Techniques

An algorithm defines the systematic steps that are needed to do some computation. Use of algorithms in sequence databases is the dominant application of string algorithms molecular biology. With all tools available to compare two strings at a time, when one has a query string and wants to find similar strings in database, why not just align the query string in turn to each of the string in the database and then report the best alignment. Further more one can use advanced methods of dynamic programming to speed up the computations and reduce the space consumption. It is claimed that the day will come when there will be little additional algorithmic detail to discuss. The key is that although databases are getting larger, proteins are not. The size of query and database strings will rarely be larger than 500 and will be usually less. Therefore the computation time will only grow linearly as the number of strings in database increases. Given the increase in computer in computer speed the decline in processors cost, and ability to trivially divide up a database search among many processors one can see a day when it may be practical to search the database by repeatedly optimizing a precise objective function using some version of dynamic programming. Even today services are available to search sequence databases using specialized chips and other hardware to implement sequence alignment very rapidly.

The issues other than computer efficiency also dictate the use of strategies like dynamic programming. Many refinements have been done in dynamic programming to speed up searching with alignment of two strings by reducing space requirements. Two strings having length n, m will require O(nm) space. But limiting resource in string alignment is not time but space. These limits makes to difficult to handle large strings no matter how long we may be willing to wait for the computation to finish. Therefore, here is the need for the method that will reduce required space from O(nm) to O(n) for (n<m).

3.1 Less memory

The similarity of two strings is a number, and that under the similarity objective function there is an optimal alignment whose value equals that number. If we only require V(n, m), and not an actual alignment with that value then the maximum space needed (in addition to the space for strings) can be reduced to 2m.For computing the value of V(n,m) we need a matrix which is used in many dynamic programming techniques. The idea is that when computing V values for row i, the only values needed from previous rows are from row i-1, any rows before i-1 can be discarded. This observation is clear from the recurrences for similarity. Thus we can implement dynamic programming solution using only two rows, one called row C, for current and one called row P, for previous. In each iteration row C is computed using row P, the recurrences, and the two strings. When that row C is completely filled in, the values in row P are now longer needed and C gets copied to P to prepare for the next iteration. After n iterations row C holds the value holds the value for row n, of the full table and hence V(n,m) is located in the last cell of that row. In this way V(n,m) can be computed in O(m) space and O(nm) time. Infact any single row of the table of the full table can be found and can be stored in those same time and space bounds

When looking for importance of database search in more complex applications, a more sensitive, selective and time- efficient search is needed. But when looking for highly similar sequences, one does not initially attempt to compute optimal similarity between new sequence and each database entry. Rather one runs the approximate methods like BLAST.

3.2 Heuristic Approach

It is basic local alignment search tool. BLAST has become the dominant searching engine for biological databases. The initial reason for its success is the speed, the fact that it outputs a range of solutions, and that each match is accompanied by an estimate of statistical significance. BLAST concentrates on finding regions of high local similarity in alignments without gaps, evaluated by an alphabet weight-scoring matrix. Very often in biological applications it is not sufficient to find just a single pair of sub strings of input strings of two strings S1, S2 with the optimal local alignment. Rather what is required is to find all or many pairs of sub strings that have similarity above the threshold. It is simply point out here that in practice the dynamic programming table used to solve the local suffix alignment problem is often used to find additional pairs of sub strings with high similarity.

3.3 Effectiveness of BLAST

Effectiveness of algorithm depends on the choice of scoring matrix, word size w, and threshold value t. lowering t reduces the chances that a sequence with an MSP score above drop off will be missed but increases the amount of computation required. By increasing the word size program becomes faster. But lower word size increases the sensitivity of BLAST. Choice of scoring matrix used can have a large effect on search results. It is sometimes suggested that the proper scoring matrix is the most critical technical element in a successful search of protein database. Ideally the score in matrix should reflect the biological phenomena that the alignment seeks to expose. The first known major amino acid substitution matrix is PAM. The term PAM that is an acronym for “point accepted mutation” or “percent accepted mutations” has two related uses. First, it is used as a unit to measure of the amount of evolutionary divergence between two amino acid sequences. Second the term PAM is used to refer to certain amino acid substitution matrices whose score have a relationship to PAM units. These choices of w, t, and scoring matrix have been studied empirically, and the default has changed over time. These empirical results will continue to vary as the methods, data, and biological concerns vary. Mainly there is need of good scoring matrices. Improvements have to make in scoring matrices. And in selection of proper values of w r t. some selection parameters also contribute towards sensitivity. So there is a need to give a strategy that will include all the issues discussed above. Existing parameters of BLAST has to be improved so that it can cope up with the new growing size of the databases. So that searches can become more fast and efficient. Sequences with much similarity can be found easily with less computation time and space.

4. Other issues

It is desirable to make database searching as much of push button exercise as possible, without requiring great expertise from user. But today effective database searching often requires a judicious mix of biological and statistical insights. The major technical improvement in protein sequence comparison over the past years has been the incorporation of statistical estimates in widely used similarity searching programs. Better estimates of statistical significance have led to improved sensitivity and selectivity and therefore to more effective search programs. These estimates are due to theorems on the statistics of MSPs. The searches, which are significant, have higher score as compared to which are not required.

Second major non-algorithmic issue of database search is “Importance Of Searching Protein With Protein”. Overwhelmingly toady, new protein sequences are obtained by sequencing the underlying DNA in the gene that codes for the protein. Therefore, most of the entries in the protein database are actually derived amino acid sequences and their originating DNA sequences are contained in DNA database. So when a searcher wants to search for a similar protein sequence the searcher can often use either the DNA sequence of the query protein to search the DNA databases or the translated sequence to search the protein databases. The reason to translate is that more sensitive and informative comparison is possible between amino acid sequences than are possible between DNA sequences. Derived amino acid strings allow more meaningful alignments by using scoring matrices that reflect the evolutionary, biological or chemical similarities of specific amino acid pairs.

5. Conclusion

Most important application is sequence alignment and comparison. This is only possible with the help of effective search algorithms. Related to algorithms there are issues like space and time reduction, which are critical, algorithmic. As continuously size is getting increased effective ways are needs to improve theses algorithms. Various parameters of these algorithms need to get improved time to time. In this BLAST algorithm has been disscussed along with the parameters affecting the search. An improvement needs to be done to improve the selection of these parameters so make search fast.

References

[1] D. Gusfield, Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology, Cambridge University Press, New York, 1997, pp. 62– 63.

[2] L. Parida, A. Floratos, and I. Rigoutsos, "An Approximation Algorithm for Alignment of Multiple Sequences Using Motif Discovery," *Journal of Combinatorial Optimization* **3**, 247–275 (1999).

[3] S. Altschul, M. Boguski, W. Gish, and J. C. Wooton, "Issues in Searching Molecular Sequence Databases,"Nature Genet. **6,** 119 –129 (1992).