Drawing Dot-matrix similarity plots with the sequence position tree.
Christophe Lefèvre. (Genosphere Project, ERATO JRDC).

Abstract

The representation of sequence similarity by dot matrix plots is a widely used method for comparing two biological sequences. It presents the user an over-all view of similarity between two sequences. We reconsider the computation of this plot. Improvement is proposed through the preprocessing of the sequence into a an automaton recognizing the word structure of a sequence. The main advantage of this approach is to eliminate systematically the repetitions during word comparison. As a result, large sequences can be handled more efficiently.

Introduction

The visual representation of sequence relatedness with dot-matrix similarity plots is one direct way to compare biological sequences. The principle is very simple. For two sequences of length N and M, a dot matrix of N rows and M columns is computed. A ‘dot’ indicates a match between the letter, word or pattern at the positions corresponding to the row and column in the respective sequences. Extended regions of similarity tend to form diagonals in the plot and, the pattern recognition abilities of the human brain allows the user to identify those regions instantly. This technique is rather subjective and only semiquantitative but presents an overall picture of sequence similarity in the data. As the biological interests in sequence similarity involves many different molecular mechanisms, the molecular biologist sometimes favors this suggestive approach over more rigorous similarity searches that may overlook some important local features. This aspect is particularly relevant to the consideration of larger sequences such as DNA.

Because the evaluation of dot matrix calls on the natural pattern recognition capacity and the ‘subjective’ knowledge of the biological expert, we reconsidered the problem of the computation of dot matrix similarity plots in ‘minimal’ time in order to achieve ‘maximum’ interactivity between the expert and the data. The basic algorithm to compute dot-matrix (ref.1) can be improved through the use of an oligomer table (ref.2). Here, further improvement is proposed through the compilation of a data structure; the position end-tree tree, which can be viewed as an automaton recognizing the word structure of a sequence.
Method

The space economical position end-set tree, briefly presented in figure 1, is constructed from the sequence data during a preprocessing step. An algorithm for its compilation in time linear with the length of the sequence data is described elsewhere (ref.3). It requires about 1 MByte of memory for every 27 kbases of DNA. This index to the sequence provides rapid string searching for the extensive localization of words or patterns in a main text (ref.4). The use of the position tree avoids the problem of extensive repetitions during the consideration of words that appears many times in the data. The algorithm compares the position tree of one sequence with the tree of the other. When a word match is found, the position of occurrence of the word are obtained from the sub-tree rooted at the corresponding node. Patterns are considered as words containing substitutions. The pattern matching algorithm explores the position tree and considers, without repetitions, increasing prefix length of all words in a sequence to recognize the matches from the position tree of the other sequence. Of course, execution time increases rapidly with the size of the sequence data and the number of substitutions permitted. One way to place constraints upon this extensive pattern searching is to implement restrictions on the distribution of substitutions over the length of the words. The user may decide that n\textsuperscript{th} mismatch may not be permitted before a match (with n-1 substitutions) of minimal length. This kind of heuristics presents a double advantage: first, the execution time is reduced dramatically as the number of prefixes that have to be taken into consideration is reduced; second, this process may act as a filtering and reduce the amount of noise in the dot plot.

**Figure 1**  Construction of a compact lexical tree for word recognition in the sequence: "taataa$".

A: Position tree

B: End label Position tree

C: Position End-set tree

In A the classic position tree is represented. The leaves of the tree are labelled with the starting positions of the word composing the path from the root to the corresponding leaf. In B the leaves of the position tree are labelled with the ending positions rather than the starting positions of the corresponding words. Several branches become equivalent upon end position labelling, as illustrated in the figure. Equivalent nodes can be grouped to produce the compact position end-set tree as represented in C.
Applications

Figure 2 shows two dot matrix obtained when comparing the human and the rat growth hormone gene (global similarity ~49%). 3a) perfectly conserved 10-nucleotides long tuple, 3b) 30-nucleotides long patterns containing at most 10 mismatches with no more than 1 mismatch for every 2 letters. Computation on a macintosh Ilfx takes about 1 and 4 seconds. Diagonals identify conserved regions corresponding to the promoter and coding regions.

Figure 2.
Dot matrix of the human (left) and rat (top) growth hormone gene

Figure 3 shows the dot matrix obtained for a 65 kbases human genome fragment encoding the growth hormone gene in a cluster containing a total of 5 related genes that emerged recently through a series of duplication and gene conversion events(ref.5). In figure 3a this sequence is compare to itself using perfect matches of 20-nucleotides long words and is computed in 37 seconds, extended diagonals indicate large regions of similarities. In figure 3b the sequence is compare to its reverse complement and takes only 2 seconds to compute. The comparison of these computing times illustrate the fact that the program spends most of its time during the identification of similarity when there is some.

Figure 3.
Dot matrix of the human growth hormone gene cluster
Figure 4 shows a matrix obtained with the complete genome of the phage lambda and illustrate the use of dot matrix for the study of genomic sequence structure. The plot was computed in about 30 seconds for perfectly conserved 7-nucleotides long words. It contains a large density of dots. However, the upper left area is of darker intensity that the rest of the plot. This observation suggest that the word composition of this genome is not homogenous. The left part of this sequence corresponds to the late genes encoding the head proteins. Thus, the use of dot matrix for the analysis of complete genomes may allows the identification of peculiar sequence features revealing genome organization.

Figure 4.

Dot plot of the phage lambda

conclusion

An algorithm for the computation of dot matrix similarity plots was developed around the use of an automaton recognizing the word structure of a sequence. This leads to some improvement regarding the computing time require for this process. Large sequences can be handled in reasonable time. We illustrate the use of this program for the analysis of sequence similarity and suggest its use for the study of genome sequence structure. DNA sequencing technology tends toward the generation of larger sequence data. The program presented here was implemented on the macintosh and it may fulfill a need for more efficient computational tools for the analysis of this growing amount of data.

3 & 4. Lefèvre, c. CABIOS, in press.