A Fast Algorithm For Sequence Clustering

Ketil Malde, Eivind Coward, and Inge Jonassen

August 3, 2002

Introduction

Expressed sequence tags (ESTs) are a valuable and rapidly growing resource for different kinds of gene analysis; for instance identification of genes, analysis of gene structure (including alternative splicing), and gene expression analysis.

Because of high redundancy, low quality, and short sequences, clustering of related EST sequences is important in order to extract useful information efficiently.

Most contemporary methods for sequence clustering (Holm and Sander, 1998; Burke et al., 1999; Liang et al., 2000) calculate a similarity (or distance) matrix, containing for each sequence a score indicating the similarity to all the other sequences. The similarity score can either be calculated using standard software like BLAST or FASTA, or use other approaches, like d2_cluster's word-based approach (Burke et al., 1999).

While effective, a direct calculation of the score matrix imposes $O(m^2)$ time and space complexity (where m is the number of sequences to be clustered).

The algorithm described in this paper uses a different approach, inspired by suffix trees, to perform EST clustering in close to linear time.

Algorithm

The algorithm is divided into three parts. The first part of the algorithm attempts to identify the pairs that have matching blocks. The second part uses the information generated from this process to calculate a score for pairs of sequences, and finally these scores form the basis for a hierarchical clustering.

1. Identify all matching blocks of length k:
   (a) Construct all suffixes from the data
   (b) Sort the suffixes into a suffix list
   (c) Generate cliques of sequences that share a matching block of length at least k
   (d) For each clique, generate all pairs of sequences
2. Score the resulting sequence pairs:

   (a) For each pair of sequences sharing a matching block, collect all other matching blocks between the sequences
   (b) Calculate the largest consistent set of matching blocks, and the score for each pair

3. Generate the clustering:
   (a) Starting with the highest scoring pair and working downward, build clusters by connecting sequences

Results and conclusion

The algorithm was implemented in Haskell (http://www.haskell.org), a high level functional language, and the protoype has been run on a test data set consisting of 10000 ESTs extracted from human eye tissue.

For the test data, the program runs faster than other contemporary clustering programs, and displays good asymptotic scalability. The resulting clusterings for different choices of parameters have been compared to other clusterings with satisfactory results, using both the Jaccard index, and a count of exactly matching clusters.

References

Burke, J., Davidson, D., and Hide, W. (1999). d2_cluster: A validated method for clustering EST and full-length cDNA sequences. *Genome Research*, 9:1135–1142.

Holm, L. and Sander, C. (1998). Removing near-neighbour redundancy from large protein sequence collections. *Bioinformatics*, 14(5):423–429.

Liang, F., Holt, I., Pertea, G., Karamycheva, S., Salzberg, S. L., and Quackenbush, J. (2000). An optimized protocol for analysis of EST sequences. *Nucleic Acids Research Vol 28 No 18*, pages 3657–3665.