A Fast Shotgun Assembly Heuristic

Christopher Wilks
San Jose State University
rew9@pacbell.net

Sami Khuri
San Jose State University
khuri@cs.sjsu.edu

Abstract

Genome sequencing opened a new era in genetics allowing the study of genomes at the nucleotide level. However, the chosen method of sequencing produced large numbers of nucleotide fragments which had to be re-assembled. The re-assembly of string fragments is known to be NP-hard. We report the results of our fast heuristic implementation for reassembling DNA fragments based on a unique approach to the problem called, "A Structured Pattern Matching Approach to Shotgun Sequence Assembly," (AMASS) created by Sun Kim. The algorithm's main idea is taken from the biological concept of probe hybridization where certain strands of nucleic acids are identified by short, unique sequences of bases that are contained within much longer DNA strands.

1. Introduction

Genome sizes are incredibly long and require complex processes to manipulate and extract meaningful information. Sequencing is the process of determining the exact ordering of the four deoxyribonucleic acid bases (DNA) in a particular genomic strand. Despite the progress made in many areas of the science and technology relating to biological studies, we still cannot sequence DNA strands whose length is above 1000 base pairs. This problem was ultimately overcome using the now famous shotgun sequencing method to accurately deduce the order of base pairs that make up the human genome [4].

Shotgun assembly can produce a relatively large number of DNA fragments that need to be re-assembled before the sequencing can be completed. Because the problem of re-assembly is NP-hard approximation and heuristic methods have to be employed. The AMASS algorithm is one such heuristic that takes a fairly unique approach to re-assembling DNA fragments [4].

2. The AMASS Heuristic

In the first phase of the algorithm, a set number of fixed length probes are randomly chosen from each fragment in the input set [4]. These probes are then converted to binary using a simple bit encoding scheme [3]. The converted probes are stored in a hash table keyed by a binary mask of their base pairs and compared against every fragment to determine all the occurrences of probes on all fragments [3]. The resulting set of fragments represented by probe occurrences and positions instead of the full nucleotide sequence makes comparison between fragments less performance than direct sequence alignment.

2.1. Benefits of Probe Matching

By encoding the probes into a binary representation and hashing them using a bit mask, the task of detecting which probes fall on which fragments is very efficient [4]. This is a much faster method than computing sequence alignments, although possibly at the cost of a decrease in sensitivity.

Comparison between probes is required for determining the overlap map of all the fragments in the set, which will in turn lead to a partial reconstruction of the target sequence [4]. Once this overlap map has been constructed, fragments can be collected into contigs, which are groups of fragments with high overlap scores.

2.2. Contig Ordering

Ideally, there would only be one contig at the end of the process, but such is not the case. Minimizing the number of contigs is an important goal, as they will later have to be manually edited to find the true, original target sequence [4]. Ordering these contigs is an imperfect process at this point and requires additional work.

3. Testing and Comparison

Our implementation was done in Java and tested on eleven genomic sequences mostly drawn from GenBank [1,7]. We analyzed the running time of the algorithm and compared our results to those obtained by an implementation of the Greedy approach to the Shortest Common Superstring abstraction of the fragment assembly problem also coded in Java [5,6]. The comparison demonstrated that the use of an approximation approach such as this one is superior in both speed and the quality of the re-assembly.

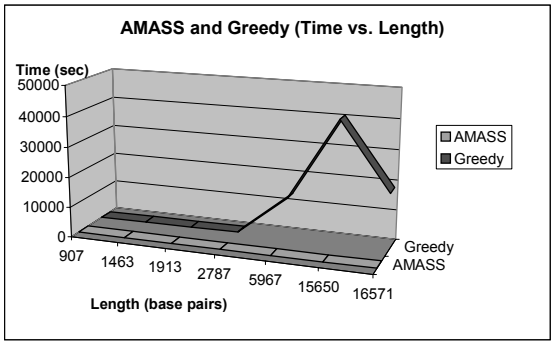

Figure 1.
Length vs. running time comparison

Another element of the testing (although not graphed) was measuring the original sequence length versus the output sequence length [5]. In a perfect world the two would be the same. However, due to certain time and knowledge constraints and the number of errors that are associated with re-assembly, we had to settle for a multiplicative distance factor, which was roughly two times the length of the original sequence.

4. Recent Development

Further updates were made to our implementation after the initial results were produced. These updates diverged from the original author's outline for comparing fragments in the initial stage of the overlap map construction. These changes resulted in a dramatic decrease in the running time for most sequences re-tested with the new version at the cost of a relatively small increase in output sequence length.

5. Future Development

Potential enhancements that could be made to our implementation include adding method(s) to mask out redundant fragments and contigs, extending the contig to fragment matching code to support faster comparisons, and the inclusion of a more robust method of ordering the contigs before reconstructing the target sequence.

6. References

[1] Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL. (2004). GenBank: update. National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health.Nucleic Acids Res. 2004 Jan 1;32 Database issue: D23-6.
http://www.ncbi.nlm.nih.gov/

[2] Kim, S. (1997); A Structured Pattern Matching Approach to Shotgun Sequence Assembly. PhD Dissertation, Computer Science Department, The University of Iowa, Iowa City.

[3] Kim, S., Kim, Y. (1998); A Fast Multiple String-Pattern Matching Algorithm. Proc. 17th Aom/IaoM Conference on ComputerScience
http://bio.informatics.indiana.edu/sunkim/STRING/

[4] Kim S. & Segre A. M. (1999); AMASS: a structured pattern matching approach to shotgun sequence assembly.
J Comput Biol, 6(2):163-86

[5] Kirchherr, Walter. (2003). CS123A,B Bioinformatics 1 & 2. Computer Science Department,
San Jose State University

[6] Meidanis, Joao., Setubal, Joao. (1997). Introduction to Computational Molecular Biology. University of Campinas, Brazil. PWS Publishing Company.

[7] Risk, Martin., Wilks, Chris., (2003). Java Implementation of AMASS Algorithm for DNA sequence Fragment Reassembly.