



DATA COMPRESSION USE IN GENOME SEQUENCE

Md. Syed Mahamud Hossein

Haldia Institute Of Technology

Neha Vinayak

Haldia Institute Of Technology

Moumita Bhunia

Haldia Institute Of Technology

Payel Chatterjee

Haldia Institute Of Technology

Anwesha Ghorai

Haldia Institute Of Technology

ABSTRACT

More and more DNA and RNA sequences are becoming available day by day. Storing and transmitting them may require a huge amount of space. Compression of these genome sequences will help us increase the efficiency of their use. We use the Huffman coding algorithm, implemented in C, to compress the given genome sequence. To design the Graphical User Interface we use Java-Swing.

KEYWORDS : DNA, Compression

INTRODUCTION

Biological sequence compression is a useful tool to recover information from biological sequences. Better compression implies better understanding. Today, the complete DNA sequences of many organisms are already known, and the completion of human genome project is making steady progress. The information of DNA sequences, RNA sequences and amino-acid sequences of proteins are stored in molecular biology databases. It is well known that the sizes of these databases are increasing nowadays very fast. Therefore it is needed to store and communicate data efficiently.

Since DNA sequences contain four symbols 'a', 't', 'g', and 'c', if these were totally random, the most efficient way to represent them would be using two bits for each symbol. However, only a small fraction of DNA sequences result in a viable organism, therefore the sequences which appear in a living organism are expected to be nonrandom and have some constraints. In other words, they should be compressible.

METHOD

We use Huffman coding to encrypt the nucleotide genome sequence, a greedy algorithm that constructs an optimal prefix code called a Huffman code. The algorithm builds the tree T corresponding to the optimal code in a bottom-up manner. It begins with a set of $|c|$ leaves and perform $|c|-1$ "merging" operations to create the final tree.

Data Structure used: Priority queue = Q

Huffman I

$n = |c|$

$Q = c$

for $i = 1$ to $n-1$

do $z = \text{Allocate-Node}()$

$x = \text{left}[z] = \text{EXTRACT_MIN}(Q)$

$y = \text{right}[z] = \text{EXTRACT_MIN}(Q)$

$f[z] = f[x] + f[y]$

INSERT (Q, z)

return EXTRACT_MIN(Q)

The total running time of Huffman on the set of n characters is $O(n \lg n)$.

MATERIALS**Sweet Potato Virus**

l tcaggcactg aaagaacaaa agacgtgga accccaacac
cagcaaaac agttaagaca

6l agaacaggac aaactcaacc gcttaagca ccagaaggga
gcacggatcc aacagatcca

12l ccactccaa cagtgaaga gataattgaa gaagaacac
cagcacaaaa agcattgagg
18l gaagcccggt gcaagcaacc agcaacacaa cctcatata
cttagggcg agacacagga
24l ccacgtagcc caagcaagt cacaacaaca agtggagtta
gggtagagatgttaatgct
30l ggaacagtag ggacgtttac agttccaaga ctccaataca
catcaagcaa gaagagatta
36l ccaatagttg acggacgtcc agtaatacaac ctggatcaact tggcagttta
cgatccagag
42l caaacaatac ttgcaatac cagatcaaca caagaacagt
ttaaggcatg gtagaagg
48l gtgaagggtg attatggggt atctgatgct gaaatgggca tactcctaa
tggcctcatg
54l gtttggtgta ttgagaatgg tacatcacccg aatattaatg gaatgtgggt
gatgatggac
60l ggagaagaac aagtaactta tccaataaaa cctctattgg
atcatgtctg cccacattt
66l agacagataa tgacacactt cagcgacata gctgaagcgt
acattgaaaa gagaacagg
72l ataaaggcct atatgccaag gtagggccta caggggaatt
tgactgatag gtagtctg
78l cggtatgcat ttgatttcta tgaactccac tcaaacacac cagtaagagc
aagggaggga
84l catatgcaaa tgaagcagc agctttaaag aatgcacaga
accgctgtt gggttggt
90l ggaacgtct ccacgaaga agaagacacg gagaggcata
caacaactga tgttacaaga
96l aatatacata acctgttagg aatgagaggt gtgcagtaaa
caatatattg ctgctacct
102l taatttcagt tggcttttaa tttaattcg tgtcttcag tccgaagag
tgttggtgt
108l ggtgtagtaac tatgtgtggt tgtaccacccg ttgctacata
taagaaaacc tcttttatt
114l acgtatcata agggactctt aaaagtgagt ctttgactcg
taagaaaagc ctttttggt
120l cgtgatcgag aa

Potato spindle tuber viroid

l cggaactaaa cctgtggttc ctgtggttca caactgacct cctgagcaga
aaagaaaaaa

6l gaatggcggc tcggaggagc gcttcaggga tccccgggga
aacctggagc gaactggcaa

12l aaaaggacgg tggggagtgc ccagcggccg acaggagtaa
ttcccgcaga aacagggttt

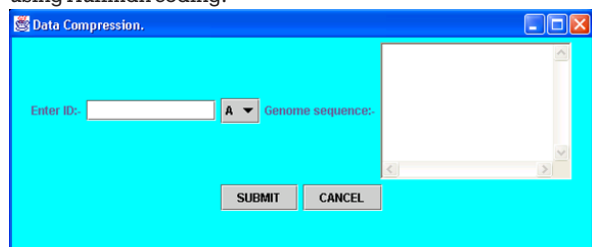
18l tcacccttc ttcttcggg tgccttct cgcgcccgca ggtccaccc
tcgccctt

24l tgcgtgtgct cttcggttac taccgggtgg aaacaactga
agctcccgag aaccgcttt

301 tctctatctt ccttgctcg gggcgagggt gtttagcct tggaccgca
gttggttct

RESULT

The GUI interface is designed using Java Swing. It takes the ID of the genome sequence as input, displays the corresponding genome sequence and on submitting, it encodes the sequence using Huffman coding.



DISCUSSION

Data compression aims at encrypting a nucleotide genome sequence at the source so that we can protect the nucleotide genome sequence from unauthorized use. If the genome sequence is transmitted to the destination and an unauthorized user is able to access the sequence and make changes in it introducing some garbage information, the structure of the genome will totally change and this will lead to loss of potentiality if the genome sequence.

The second most important point we have to keep in mind is Data Compression. The sequence is compressed by means of encryption, but we have to keep in mind that it does not lead to data loss.

Future Work

We have successfully been able to encrypt the genome sequences using Huffman coding. In future we plan to use the information stored in the genome sequence, to omit sending the unwanted data in the sequence to achieve data compression without data loss.

REFERENCES

- [1]. Bennett,C., Li,M. and Ma,B. (2000) Linking chain letters. *ScientificAmerican*, to appear Benedetto,D., Caglioti,E. and Loreto,V. (2002) Language trees and zipping. *Phys. Rev. Lett.*, 88, 048702.
- [2]. Chen, X., Kwong, S., and Li, M., A compression algorithm for DNA sequences and its applications in genome comparison, *Genome Informatics*, 10:52-61, December 1999.
- [3]. Chen,X., Kwong,S. and Li,M. (2001) A compression algorithm for DNA sequences. *IEEE Engineering in Medicine and Biology Magazine*, 20, 61-66.
- [4]. Cleary, J.G. and Witten, I.H., Data compression using adaptive coding and partial string matching, *IEEE Trans. on Commun.*, COM-32(4):396-402, April 1984.
- [7]. Nevill-Manning, C.G. and Witten, I.H., Protein is incompressible, *Proc. of IEEE Data Compression Conference*, 257-266, 1999.