A New GC based HMM Algorithm for Disease Classification

Dr.V.Anuradha S.K.M.Habeeb A.Praveena Amala Priya
Department of Bioinformatics,Guru Nanak College
Velachery, Chennai, India

ABSTRACT

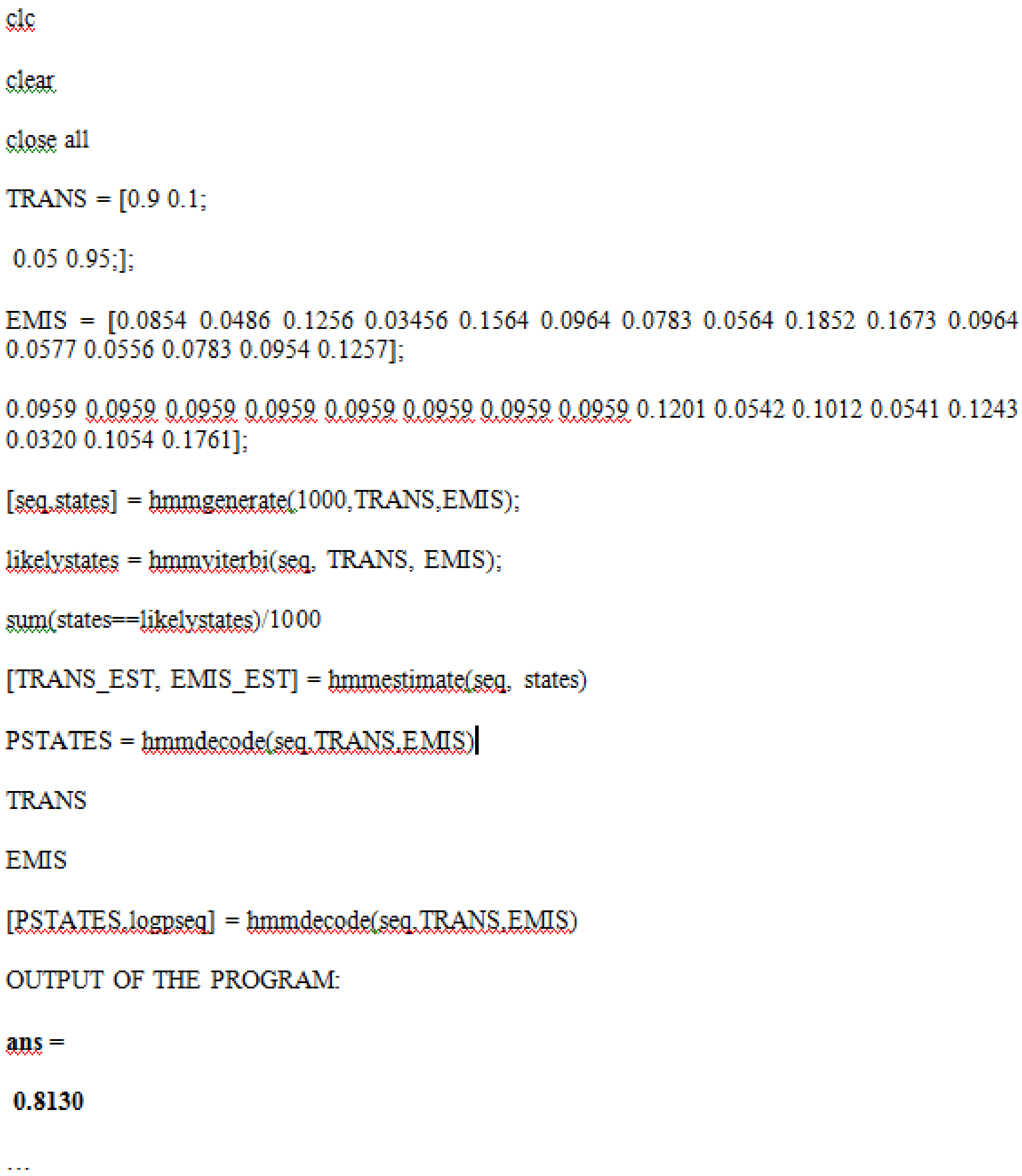
This paper presents a hidden markov model which classifies proteins into classes: the normal protein and the diseased proteins. Using a dataset of 50 protein sequences, the method was able to classify the proteins with a better accuracy of 81%. We used the HMM based software called Matlab to train the data. Matlab uses some of the HMM functions to classify the normal and diseased proteins based with the 16 combinations of amino acids. First the patterns are extracted using 2-gram amino acid encoding method. Here we have 16 patterns which codes for GC. Then scores of these 16 patterns are given as an input for hidden markov model. The hidden markov model was trained on two classes of the proteins based on the known patterns and the trained model was used to classify the dataset. Therefore, the method was able to classify the proteins with an accuracy of 81%. The results of this algorithm provide insights that can help biologists and computer scientists design high-performance protein classification systems of high quality.

Keywords: HMM, Matlab, 2-gram

1. INTRODUCTION

Machine learning is learning, like intelligence whose definition includes phrases such as “to gain knowledge, or understanding of, or skill in, by study, instructions, or experience”. It is a natural outgrowth of the intersection of computer science and statistics. It covers a broad range of learning tasks, such as how to design autonomous mobile robots that learns to navigate from its own experience, how to mine historical medical records to learn which future patients will respond best to which treatments and how to build search engines that automatically customize to their user’s interests. To be more precise, we say that a machine learns with respect to a particular task T, performance metric P, and type of experience E, if the system reliably improves its performance P at task T, following experience E (1).

The four bases found in DNA are adenine (A), cytosine (C), guanine (G) and thymine (T). Here we focus on Guanine and Cytosine content (GC). DNA with high GC-content is more stable than DNA with low GC-content. Since the protein sequence has important characteristic features for several mechanisms we focused only the four amino acids which is directly coded by G& C (2). From the universal codon table, we inferred that the A, G, P, R amino acids are directly coded by G & C (3). G and C directly codes for four amino acids in proteins namely A, P, G and R. Using 2-gram encoding method, the 16 combinations of these amino acids are taken as an input Hidden Markov Model. GC-content (guanine-cytosine content) is a characteristic of the genome of any given organism or any other piece of DNA. The variations in GC ratio within a genome of higher organism’s results in a mosaic like formation with islet regions called isochores (4). The isochores include in them are essential protein coding genes, termed house keeping genes and thus determination of ratio of these specific regions contributes in mapping these essential genes (5). Evidence of GC ratio with that of length of the coding region of a gene have showed that the length of the coding sequence is directly proportional to higher G+C content (6). It has been used to scan the basic makeup of the genome, as well as understanding coding sequence evolution (7). Sharp changes in GC-content are detected at the transcription boundaries for all species analyzed, including human, mouse, rat, chicken, fruit fly and worm. In vertebrates, sharp positive and negative spike of GC-content are observed at the transcription start and stop sites respectively, and there is also a progressive decrease in GC-content from the 5’ untranslated region to the 3’ untranslated region along the gene. In invertebrates, the positive and negative GC-content spike at the transcription start and stop sites are preceded by spikes of opposite value and the highest GC-content is found in the coding regions of the genes. It may reflect a general principle of genomic punctuation (8). It is already a unique feature for in silico gene identification (9). Regions with sharp GC-content changes should be structurally different from the rest of the genome, where the GC-content is constant or gradually changing and thus should be recognizable from the rest of the genome. Therefore, GC-content spikes may play a general role in delineating different functional regions of



```
clc

clear

close all

TRANS = [0.9 0.1;

 0.05 0.95;];

EMIS = [0.0854 0.0486 0.1256 0.03456 0.1564 0.0964 0.0783 0.0564 0.1852 0.1673 0.0964
0.0577 0.0556 0.0783 0.0954 0.1257];

0.0959 0.0959 0.0959 0.0959 0.0959 0.0959 0.0959 0.0959 0.1201 0.0542 0.1012 0.0541 0.1243
0.0320 0.1054 0.1761];

[seq,states] = hmmgenerate(1000,TRANS,EMIS);

likelystates = hmmviterbi(seq, TRANS, EMIS);

sum(states==likelystates)/1000

[TRANS_EST, EMIS_EST] = hmmestimate(seq, states)

PSTATES = hmmdecode(seq,TRANS,EMIS)

TRANS

EMIS

[PSTATES,logpseq] = hmmdecode(seq,TRANS,EMIS)

OUTPUT OF THE PROGRAM:

ans =

 0.8130

...
```

Figure:1. Matlab code and truncated output showing the value 0.8130 indicating the accuracy of 81%.

the genome (10).Translation initiation sites were also characterized by sharp GC-content spikes (11). In our study here we focus on the 2-gram amino acids coded by these multi-faceted GC rich codons using the Matlab's HMM concept to identify the diseased and the normal protein.

2. MATERIALS AND METHODS

2.1 HMM

A Hidden Markov model (HMM) is one of the machine learning algorithms. It is a statistical tool which allows us to model complex stochastic phenomenon. The important steps of HMM include architectute design, learning & traininig and recognition & classification. The performance levels were evaluated using statistical measures such as accuracy, sensitivity and specificity.

2.2 GENBANK

The data required for the study was obtained from the genbank database at http://www.ncbi.nlm.nih.gov/genbank.

2.3 FEATURE EXTRACTION

Here we mainly focused on GC content which directly codes for amino acids namely A, P, G and R. The feature extraction is based on the 2-gram amino acid encoding method. There are 16 combinations of 2-grams for these four amino acids. The 2-gram amino acid encoding is the method that counts the occurrences of 2 consecutive amino acids in protein sequences (12). To calculate the global similarity of protein sequences, we adopt the 2-gram, also known as 2-tuple, method as described in Wu. The 2-gram encoding method extracts various patterns of two consecutive amino acid residues in a protein sequence and counts the number of occurrences of the extracted residue pairs. In case of protein sequences (20 amino acids), there are 400 possible 2-grams, that produce a large feature space.

2.4 PERL SCRIPT

A Perl script was written to count the no. of 2-gram patterns in the given protein dataset. In order to convert these protein sequences into scores, a Perl script has been written which gives the scores based on the 16 combination of 2-grams (Appendix 1). The resulting count obtained from the Perl program is seeded as training set to the HMM using Matlab software.

2.5 MATLAB

MATLAB is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. It allows both “programming in the small” to rapidly create quick programs you do not intend to reuse. Statistics tool box functions such as hmmgenerate, hmmestimate, hmmtrain, hmmviterbi and hmmdecode were used in our study.

3. RESULTS AND DISCUSSIONS

The results should show the performance of algorithm in terms of accuracy, sensitivity and specificity. It will be based on the training which has been given. In the above program, transition probabilities for normal and diseased proteins were assumed as 0.9, 0.1 and 0.05, 0.95 shown in figure.1. The first one denotes that the normal protein can have a 90% of chance to convert into a diseased protein and the diseased proteins have only 10% of chance to be converted to a normal protein. And the second one shows that the diseased proteins have a 95% chance of getting converted into a normal protein and the normal proteins have only 5% chance of getting converted to diseased proteins. Therefore, the sum of all transition probability equals 1. The emission probabilities were assigned based on the scores which we got as output of Perl program. The sum of all the scores for normal proteins were taken and the probability of the 16 2-gram in the normal proteins is calculated. Similarly, the probabilities in case of diseased proteins were calculated. The result obtained from this exercise showed that this strategy successfully categorized the diseased and the normal protein at the accuracy rate of 81%.

4. CONCLUSION

We have started the algorithm with the aim of getting a better accuracy to classify normal and diseased proteins using the machine learning algorithm, HMM. With the aim of getting better accuracy, input and training has been given to the HMM with 50 datasets. Using the method of 2-gram amino acid encoding, we focused only on the GC content. The input given is the scores of 16 features in the normal and diseased proteins. Therefore we can conclude that the predictions made by the HMM using amino acid encoding method achieved a better accuracy of 81.30%. Since we have used a small dataset of 50 protein sequences, the accuracy is less. The accuracy can be increased by increasing the size of the dataset. Hence, this method of classifying the normal and diseased proteins using HMM may help for the future research.

5. REFERENCES

[1] Tom M. Mitchell, 2006, The Discipline of Machine Learning.

[2] Swanson, R. 1984. A unifying concept for the amino acid code. Bull. Math. Biol., 46, 187-207.

[3] Bosnacki, D., ten Eikelder, H.M.M., Hilbers, P.A.J. Genetic Code as a Gray Code Revisited. In the Proceedings of International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences.

[4] Bernardi, G. 2000. Isochores and the evolutionary genomics of vertebrates, Gene, 241: 3-17

[5] Aïssani, B., and Bernardi, G. 1991. CpG islands, genes and isochores in the genomes of vertebrates, Gene, 106:185-195.

[6] Oliver, JL. and Marín, A. 2004. A Relationship Between GC Content and Coding-Sequence Length Journal of Molecular Evolution, 43(3)216-223.

[7] Hurst, LD. and Merchant, AR. 2001. High Guanine-Cytosine Content is Not an Adaptation to High Temperature: A Comparative Analysis amongst Prokaryotes Proceedings: Biological Sciences, 268(466) 493-497.

[8] Lingang Zhang., Simon Kasif., Charles R. Cantor and Natalia E. Broude. 2004. GC/AT-content spikes as genomic punctuation marks, 101: 48, 16855–16860.

[9] Yeramian, E and Jones L. 2003. GeneFizz: A web tool to compare genetic (coding/non-coding) and physical (helix/coil) segmentations of DNA sequences. Gene discovery and evolutionary perspectives. Nucleic Acids Res 31: 3843–3849.

[10] Vinogradov, A. E. 2001. Mol. Biol. Evol. 18**,** 2195–2200.

[11]Mizuno, M., and Kanehisa, M. 1994. Distribution profiles of GC content aroundthe translation initiation site in different species. FEBS Lett 352, 7-10.

[12] Wang, J. T. L., Ma, Q., Shasha, D., and Wu, C. H. 2001. New techniques for extracting features from protein sequences. *IBM:* Systems Journal, 40(2):426–441.

APPENDIX. 1

CONVERSION OF PROTEIN SEQUENCES INTO SCORES:

```
#! perl/bin/perl –w
print "Please type the filename of the feature data: \n\n";
$patternfilename = <STDIN>;
chomp $ patternfilename;
unless (open(PATTERNFILE, $patternfilename)) {
  print "cannot open file\"$ patternfilename\"\n\n";
  exit;
  }
@ pattern = <PATTERNFILE>;
close PATTERNFILE;
$ pattern = join( ", @ pattern);
$ pattern =~ s/\s//g;
$lengthseq = length($pattern);
print "\nLength of the sequence is : $lengthseq \n\n";
$aa = 0; $ag = 0; $ap = 0; $ar = 0; $ga = 0; $gg = 0; $gp = 0;
$gr = 0; $pa = 0; $pg = 0; $pp = 0; $pr = 0; $ra = 0; $rg = 0;
$rp = 0; $rr = 0; $ee = 0;
while($pattern =~ /aa/ig){++$aa}
while($pattern =~ /ag/ig){++$ag}
while($pattern =~ /ap/ig){++$ap}
while($pattern =~ /ar/ig){++$ar}
while($pattern =~ /ga/ig){++$ga}
while($pattern =~ /gg/ig){++$gg}
while($pattern =~ /gp/ig){++$gp}
while($pattern =~ /gr/ig){++$gr}
while($pattern =~ /pa/ig){++$pa}
while($pattern =~ /pg/ig){++$pg}
while($pattern =~ /pp/ig){++$pp}
while($pattern =~ /pr/ig){++$pr}
while($pattern =~ /ra/ig){++$ra}
while($pattern =~ /rg/ig){++$rg}
while($pattern =~ /rp/ig){++$rp}
while($pattern =~ /rr/ig){++$rr}
while($pattern =~ /[^agpr]/ig){++$ee}
print "AA = $aa  AG = $ag  AP = $ap AR = $ar \n\n";
print "GA = $ga  GG = $gg  GP = $gp  GR = $gr \n\n";
print "PA = $pa  PG = $pg  PP = $pp  PR = $pr \n\n";
print "RA =$ra  RG = $rg  RP = $rp  RR = $rr \n\n";
print "The values are : \n\n";
$x1 = ($aa/($lengthseq - 1)); print "AA = $x1 \n";
$x2= ($ag/($lengthseq - 1)); print "AG = $x2 \n";
$x3 = ($ap/($lengthseq - 1)); print "AP = $x3 \n";
$x4 = ($ar/($lengthseq - 1)); print "AR = $x4 \n";
$x5 = ($ga/($lengthseq -1)); print "GA = $x5 \n";
$x6 = ($gg/($lengthseq - 1)); print "GG = $x6 \n";
$x7 = ($gp/($lengthseq - 1)); print "GP = $x7 \n";
$x8 = ($gr/($lengthseq - 1)); print "GR = $x8 \n";
$x9 = ($pa/($lengthseq - 1)); print "PA = $x9 \n";
$x10 = ($pg/($lengthseq - 1)); print "PG = $x10 \n";
$x11 = ($pp/($lengthseq - 1)); print "PP = $x11 \n";
$x12 = ($pr/($lengthseq - 1)); print "PR = $x12 \n";
$x13= ($ra/($lengthseq - 1)); print "RA = $x13 \n";
$x14 = ($rg/($lengthseq - 1)); print "RG = $x14  \n";
$x15 = ($rp/($lengthseq - 1)); print "RP = $x15  \n";
$x16 = ($rr/($lengthseq - 1)); print "RR = $x16  \n";
$outputfile = "count2gram.txt";
unless (open(COUNT2GRAM, ">$outputfile") )
{
print "cannot open file \"$outputfile\" to write to!!\n\n";
exit;
};
print COUNT2GRAM "AA = $x1 AG = $x2  AP = $x3  AR =
$x4  GA = $x5  GG = $x6 GP = $x7  GR = $x8 PA = $x9 PG
= $x10 PP = $x11  PR = $x12  RA = $x13  RG = $x14  RP =
$x15  RR = $x16\n";
close(COUNT2GRAM);
exit;
```