Bacteria Classification Based on 16S Ribosomal Gene Using Artificial Neural Networks

SUSAN HIGASHI  
Lab. Nacional de Computação Científica  
Lab. de bioinformática  
Av. Getúlio Vargas 333, Petrópolis  
BRASIL  
susanh@lncc.br

MARIANGELA HUNGRIA  
Embrapa Soja  
Lab. de Biotecnologia dos Solos  
Av. Carlos João Strass, Londrina  
BRASIL  
hungria@cnpso.embrapa.br

MARIA A. DE O. C. BRUNETTO  
Universidade Estadual de Londrina  
Depto. de Computação  
Rod. Celso Garcia Cid 445, Londrina  
BRASIL  
angelica@uel.br

Abstract: Bioinformatics is defined as a scientific field which solves problems of molecular biology using concepts and methods from computer science. The increasing amount of biological data has promoted the development of more and more specialized systems in organizing and mining such data. One of the areas bioinformatics can help understanding is bacteria taxonomy, which consists, between other tasks, in classifying bacteria in a taxonomic level. efficiently aided by computational methods. Among them, Artificial Neural Network (ANN) and n-gram encoding method have played an important role in classifying data due to their ability to extract the essential characteristics of the data, instead of memorizing them. As a consequence, the amount of data is reduced and there is no need to create a model to the underlying problem. Based on that, we developed an ANN, which revealed itself efficient in dealing with biological data (e.g. nucletide sequences). After the ANN training process, it classified the sequences with accuracy ranging from 90% to 100%.

Key–Words: Bacteria classification, Neural Network, Nitrorgen fixation, 16S gene

1 Introduction

The term Bioinformatics came up in the beginning of 1990, when a big amount of experimental data was yielded by the sequencing projects, mainly from the Human Genome Project. Aside from the improvements in databases, the development of algorithms related to sequence alignment interested researches concerning biological processes, mainly, through the study of genome and proteome.

Considering the sequence analysis, the beginning of the bioinformatics was in 1970 with Needleman and Wunsch publication, which made the establishment of alignment algorithms feasible [3]. The fast growing amount of data coming from molecular biology projects (genomics, proteomics, transcriptomics, etc.) increased the need of developing more specialized systems in the to organise these data and to produce useful inferences about them[4, 5].

To decipher the huge and growing amount of biological data, we present some numbers: in 2004 GenBank1, the National Institute of Health (NIH2) genetic sequence database, used to maintain over 2 million sequences. Moreover, the European Molecular Biology Laboratory (EMBL3) nucleotide sequence database held in 2005 over 58.7 million entries (release 84), in 2006 there were over 80.5 million entries (release 88), which means a grow of 37.13% in just one year. Finally, the Ribosomal Database Project (RDP4) stored in 2008 around 715,637 rRNA sequences and this number grows approximately 5000 each month [8, 13, 14].

The use of sequence classification methods, specially those based on machine learning like ANN, is an alternative to similarities search methods in databases. Since the time spent by the first method mentioned grows according to the number of classes instead of the number of sequence as does the other mentioned method [4].

Developing more refined techniques to solve complex problems related to the correct interpretation of the data gathered is a growing need. This task requires databases to store the information and faster and more accurate technologies of data processing to infer how the biological processes occur. For this reason, bioinformatics has played an important role and is considered a promising field[6]. Machine learning and statistics algorithms are some of the approaches used to solve bioinformatics problems[7].

---

1GenBank is available at http://www.nlm.nih.gov/  
2NIH is available at http://www.nih.gov/  
3EMBL database is available at http://www.ebi.ac.uk/embl/  
4RDP is available at http://rdp.cme.msu.edu/
More specifically, in the agricultural field, the sustainability of food crops, forages and green manure legumes is related to their ability to establish symbiotic associations with stem and root-nodulating N₂-fixing bacteria, such as species of *Bradyrhizobium* and *Rhizobium* genus. As the taxonomic position of these species is poorly known, this work proposes an efficient approach to classify these species in taxonomic level basing on nucleotide sequences from the ribosomal gene 16S[1, 2]. The classification is based on 16S gene due to its features such as universal distribution, conservation of structure and function, ideal size (1.5 kb) making it a good molecular marker[17].

There are some approaches used to solve classification problems, like naive Bayes classifier[8], artificial neural networks[4], support vector machine, and so on. In our work, we selected Artificial Neural Networks (ANN), which has shown itself efficient and accurate in dealing with biological data[9].

The aim of this work is to present the bacteria classification problem solved by a machine learning technique, called Artificial Neural Network (ANN).

## 2 Results and Discussion

The performance evaluation of n-gram encoding method and MLP artificial neural network in the task of classifying rDNA 16S nucleotides sequences from fixing nitrogen bacteria brought good results. That is, comparing to other sequence similarity search tools, like Blast, the time spent by the encoding and classifying processes were significantly smaller and the accuracy in the sequences classification was almost the same, as shown in the table 1.

Table 1: Comparison of the ANN and Blast efficiency in the sequence classification task. The e-value, for Blast results, is the probability of the sequence alignment given by chance.

<table>
<thead>
<tr>
<th></th>
<th>Accuracy</th>
<th>e-value</th>
<th>Time spent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ANN</td>
<td>98%</td>
<td>-</td>
<td>0.28 min.</td>
</tr>
<tr>
<td>Blast</td>
<td>100%</td>
<td>0.0</td>
<td>3.37 min.</td>
</tr>
</tbody>
</table>

To represent the nucleotides sequences (*i.e.* strings) as a numerical input to the ANN, these sequences were encoded in numerical values by n-gram method. The value of the parameter \(n = 3\) was selected because a codon, which encodes an aminoacid, is comprised of three nucleotides. The encoding task were based on[4] and on[10] and the ANN development was based on [11] and on [10].

The ANN input are arrays comprised of 64 columns\(^5\), each column representing a trigram frequency (*e.g.*, trigram ATC, AAT, TTA, etc.), and \(n\) lines, each line representing a bacterium nucleotides sequence. The MLP model was developed under supervised learning paradigm, hence the desired output \(o_i\) for the \(i\)-th sequence must be informed. For that, the \(o_i\) value was included, as a column (the last one), in the input files and could assume 0 and 1 values, respectively representing the sequence as pertaining or not to the specified genus for which the ANN was developed. An example of this input array is presented in figure 5.

From the entire input data set, 1/3 was separated for test, which consists in using the best synaptics weights from the training cycles to evaluate ANN accuracy. The remaining 2/3 were used to train the network to get the best weights. The table 2 shows the total number of patterns, the one separated for training and the other one selected for testing.

Table 2: The total number of input patterns, the proportion separated for training (2/3) and for testing (1/3).

<table>
<thead>
<tr>
<th></th>
<th>Total</th>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Bradyrhizobium</em></td>
<td>65</td>
<td>43</td>
<td>22</td>
</tr>
<tr>
<td><em>Rhizobium</em></td>
<td>28</td>
<td>19</td>
<td>9</td>
</tr>
</tbody>
</table>

The bacteria were classified in the following taxonomic levels: (i) domain, (ii) phylum, (iii) class, (iv) order, (v) family and (vi) genus. The genera, *Bradyrhizobium* and *Rhizobium*, were selected because of its agronomic importance in nitrogen fixation in soybean roots. The genera lineage are: [Bacteria → Proteobacteria → Alphaproteobacteria → Rhizobiales → Bradyrhizobiaceae → Bradyrhizobium] and [Bacteria → Proteobacteria → Alphaproteobacteria → Rhizobiales → Rhizobiaceae → Rhizobium/Agrobacterium group → Rhizobium].

The following subsections summarizes the results obtained in the training processes of ANN developed for each genus.

### 2.1 Bradyrhizobium

The ANN, developed to *Bradyrhizobium*, performed an important role in this classification task. The results presented 100% accuracy in classifying sequences of *Bradyrhizobium* species and low MSE

---

\(^5\) \(m^n = 4^3 = 6\), where \(m\) is the number of elements of the alphabet (*i.e.* A, T, C and G) and \(n\) is extraction degree of n-gram.
(Mean Square Error\textsuperscript{6}). To obtain the good results, as mentioned before, the ANN was trained varying some parameters, such as epochs (\textit{ep}) number, learning rate (\(\eta\)), momentum term (\(\alpha\)) and epochs number. The best configuration was 150 to 200 epochs, with the best results obtained with \textit{epochs} = 200. The use of low learning rate values (\(\eta = 0.1\)) was not a good approach, since the ANN was not able to learn even using a great number of epochs and repetitions (\textit{rep}). On the other hand, \(\eta = 0.7\) demonstrated excellent results when combined momentum term \(\alpha = 0.5\) and a reasonable repetition number (\textit{i.e.}, \textit{repetitions} = 10). Three hidden nodes (\(hn\)) were enough to extract patterns from the sequences, which allowed ANN to classify the samples correctly. Summarizing, the best configuration was \([\textit{ep} = 200; \eta = 0.7; \alpha = 0.5; \textit{rep} = 10; \textit{hn} = 2]\) what resulted in accuracy of 100\% and \textit{MSE} = 0.001019.

2.2 Rhizobium

Similar to \textit{Bradyrhizobium} ANN, the one for \textit{Rhizobium} played its role in classifying the sequences. The ANN classified the samples with 100\% of accuracy and low MSE. The configuration \([\textit{ep} = 200; \eta = 0.7; \alpha = 0.5; \textit{rep} = 3 \text{ or } \textit{rep} = 10; \textit{hn} = 2]\) generated accuracy = 100\% and \textit{MSE} = 0.000922, which is an excellent value. Epochs ranging from 150 to 200 values were good choices. Equally to before mentioned ANN, low learning rate values (\(\eta = 0.1\)) did not produce good results, for the same reason: the ANN was not able to learn in spite of the great number of epochs and repetitions. The best value for \(\eta\) was, indeed, 0.7, which allowed excellent results jointly with momentum term \(\alpha = 0.5\). For this ANN, just 3 repetitions were enough to get good results. Again, three hidden nodes were enough to extract patterns from the sequences and allowed ANN to classify the samples correctly. Summarizing, the best configuration was \([\textit{ep} = 200; \eta = 0.7; \alpha = 0.5; \textit{rep} = 3; \textit{hn} = 2]\) what resulted in accuracy of 100\% and \textit{MSE} = 0.000922.

3 Methods

In this work, the classification of nitrogen fixing bacteria in taxonomic level is based on the linear nucleotides sequences from the ribosomal RNA 16S gene. Therefore, the ANN input data are these nucleotides sequences. In computational terms, these sequences are considered strings comprised by the alphabet A, T, C, G. The sequence acquirement and formatting are described in the following sections.

### 3.1 Input data acquirement

The 16S rDNA nucleotides sequences data were acquired from wet experiments performed at Laboratório de Biotecnologia dos Solos – Embrapa Soja.

### 3.2 Input data formatting

Among the available extensions of sequences file, such as XML, GenBank, and so on, FASTA was selected because it is simple and pragmatic. The FASTA format is made up of a header, which starts with the greater symbol (\textgreater{}), it is followed by the sequence name, optionally it is pursued by some comments and, finally, the sequence itself [12].

### 3.3 Input data encoding

The nucleotides sequences were encoded by the \(n\)-gram hashing function, which extracts and accounts the occurrences of the \textit{n} consecutives residues. ANN requires a numerical and size fixed input data. For that the \(n\)-gram method is a good choice, since it restricts the input size to \(m^n\), where \(m\) is the number of different symbols from the alphabet and \(n\) is the extraction degree required (\textit{e.g.}, 2-gram, 3-gram, ..., \(n\)-gram) [10].

For the sequences encoding task, it was developed a few Java classes aided by Netbeans 6 IDE. The encoding process (figure 1) followed three steps which were:

1. Step 1: FASTA file reading (figure 2) with Biojava package\textsuperscript{7}. The .seq output file (figure 3) generated in this process contains the terms equivalent to each nucleotide letter.

\textsuperscript{7}Biojava package is available at http://www.biojava.org

![Figure 1: Processing flow of the 16S ribosomal sequences](image)

**Figure 1:** Processing flow of the 16S ribosomal sequences. Step 1: figures 2 and 3. Step 2: figure 4. Step 3: figure 5.

**1. Step 1:** FASTA file reading (figure 2) with Biojava package\textsuperscript{7}. The .seq output file (figure 3) generated in this process contains the terms equivalent to each nucleotide letter.

\textsuperscript{6}Mean Square Error is a manner to quantify how different the result of the ANN is comparing to the real values.
2. Step 2: execution of count.pl script which accounts the frequency of each n-gram from .seq file. The count.pl output file with extension .frq (figure 4) contains the frequency of each trigram (e.g., AAA, TCC, CGG, etc.).

3. Step 3: the frequencies from .frq file were normalized between [0, 1], where the trigrams with the greatest frequency were normalized to 1, the smallest one was normalized to 0 and the remaining were normalized in rate of the greatest. Finally, the normalized values were written in a input file to the ANN, which is formatted according to the figure 5. In ANN input file, each line represents one rDNA 16S nucleotide sequence from a nitrogen fixing bacteria and the columns denotes the frequency of each $m^n$ (i.e., $m^n = 4^3 = 64$) trigrams.

---

2. Step 2: execution of count.pl script\(^8\) which accounts the frequency of each n-gram from .seq file. The count.pl output file with extension .frq (figure 4) contains the frequency of each trigram (e.g., AAA, TCC, CGG, etc.).

3. Step 3: the frequencies from .frq file were normalized between [0, 1], where the trigrams with the greatest frequency were normalized to 1, the smallest one was normalized to 0 and the remaining were normalized in rate of the greatest. Finally, the normalized values were written in a input file to the ANN, which is formatted according to the figure 5. In ANN input file, each line represents one rDNA 16S nucleotide sequence from a nitrogen fixing bacteria and the columns denotes the frequency of each $m^n$ (i.e., $m^n = 4^3 = 64$) trigrams.

---

8 count.pl script is part of Ngram Statistics Package (NSP) and is available at http://www.d.umn.edu/.

---

2. Sensitivity, the search is based on information of homologous families, which presents more information about the isolated sequences.

3. Automated families determination.

Essentially, classification/clustering systems can be based on supervised and unsupervised learning. If there is an \textit{a priori} relationship between the sequences classes to be classified, Backpropagation algorithm is a good example. The unsupervised learning (\textit{e.g.}, Self Organizing Maps (SOM)) is a good alternative when the relationship between these sequences are not known [4].

Neural Networks trained with Backpropagation represents the most used approach and are playing an important role in solving some problems, including molecular sequences classification [4]. Hence, we implemented Multi Layer Perceptron (MLP) trained with Backpropagation to classify the ribosomal 16S sequences from fixing nitrogen bacteria. The neural network development was based on [10]. The architecture of neural network was built with the following configuration:

- Input layer comprised of $m^n = 4^3 = 64$ nodes, where $m = 4$ (number of symbols from the nucleotides alphabet) and $n = 3$ (extraction degree from the n-gram method).

- An initial valuation of the hidden nodes ($hn$) number was calculated using the formula $hn = N_{trn}/[R + (N_{inp} + N_{out})]$, where $N_{trn}$ is the number of training patterns, $R \in [5; 10]$, $N_{inp}$ is
the number of input nodes and $N_{out}$ is the number of output nodes, as cited on [9]. Based on this valuation, the number of $hn$ used as parameter to the network evaluation were 2, 5, 10 and 20 hidden nodes. Evaluating the results, it was established that 5 to 10 hidden nodes were enough to extract characteristics from the training patterns. Therefore, the final ANN configuration has $hn = 10$.

- Output layer consisted of one node, since it is a binary classification. In this case, binary means: 1, if the bacteria belongs to the specified genus (i.e. *Bradyrhizobium* or *Rhizobium*) or 0, if it does not belong.

The neural network training was performed in a personal computer, with processor Intel Pentium Dual-Core (1.73 GHz) and 1Gb memory, varying five parameters:

1. Hidden nodes ($hn$): based on the initial valuation, the 2, 5, 10 and 20 values were used in each training cycle, as mentioned before.

2. Learning rate ($\eta$): the selected values were $\eta = 0.1, 0.3, 0.7$; as suggested in [9] the ideals values to $\eta$ belongs to $[0.0; 1.0]$.

3. Momentum term ($\alpha$): the selected values $\alpha = 0.0, 0.5, 1.0$ were based on [9] who cited $\alpha \in [0.0; 1]$.

4. Number of epochs ($ep$): in conformity to [9], the number of epochs is determined by the analysis of the MSE training and test curve, as can be seen on the graph from figure 6. A great number of epochs causes the memorization of patterns by the neural network, that is, the network training MSE generated is small, however, the classification results can be significantly degenerated. Initially, this number were settled according to the previously mentioned graph and, hence, the values used for each training process were 3, 10, 20 and 30 epochs.

5. Training subset size: according to cross validation method Holdout, $2/3$ from the patterns were reserved to ANN training; the remaining $1/3$ were used to test the network [15], as shown in table 2.

Finally, the total number of experiments were $hn \times \eta \times \alpha \times ep = 4 \times 3 \times 3 \times 4 = 144$.

During the ANN training, a sample was considered as correctly classified if the error $(\text{desired output} - \text{obtained output})$ was smaller than 0.2 in at least 80% of the repetitions performed. To measure the ANN performance true positive (TP), true negative (TN), false positives (FP) and false negatives (FN) were calculated to estimate the network sensitivity (equation 1) and sensibility (equation 2).

$$\text{sensitivity} = \frac{TN}{TN + FP} \times 100\% \quad (1)$$

$$\text{sensibility} = \frac{TP}{TP + FN} \times 100\% \quad (2)$$

### 4 Conclusion

The “omes” projects have been generating a huge amount of data, which encourage the development of bioinformatics tools to solve the new incoming problems. Indeed, this study is motivated by this fact.

Researches in the analysis of biological data thought computational methods are growing, however some of this methods brings uncertain results and sometimes the time to generate an output is not feasible.

Hence, the aim of this work was to organize the amount of biological (rDNA 16S sequences from nitrogen fixing bacteria) data to obtain more knowledge about the sequenced organisms and then make the bacteria classification process more efficient.

The most important factor in the bacteria classification process is the capacity to extract relevant features from the 16S gene sequences. These characteristics are related to the sequences canonical form, that is, the linear nucleotides sequences.

The neural networks are an interesting approach in the sequence classification for their capacity of generalization after a training process and their time
search is not restricted to the database size, but to the number of classes, that is a great advantage, since the number of nucleotides sequences is growing exponentially.

The encoding process, using the n-gram method, was developed with Java programing language (Biojava package) and the neural networks were implemented in Scilab. To each one of the genus, *Bradyrhizobium* and *Rhizobium*, it was developed one neural network to classify the sequence as pertaining or not to that class/genus.

The results, presented in the section 2, demonstrates that the n-gram method was able to extract important information from the sequences, which allowed the MLP model to produce high scores in the sequences classification. The good results generated by the combination of these two techniques (n-gram method and MLP neural networks) encourage the use of these approaches in solving bioinformatics problems. As a proposal for future studies is to use different neural networks models to classify the sequences. Another suggestion, is the use of others machine learning techniques, for example Bayesian networks. An alternative option to the use of neural networks is to solve different problems, such as re-construction of phylogenetics trees and searching an operon order in the genome.

Acknowledgements: The research was supported by National Counsel of Technological and Scientific Development (CNPq).

References:


