

Taxonomically Clustering Organisms Based on the Profiles of Gene Sequences Using PCA

¹E.Ramaraj and ²M.Punithavalli

¹Department of Computer Science and Engineering, Alagappa University, TN, India

²Department of Computer Science, Sri Ramakrishna College of Arts and Science for Women
Coimbatore, TN, India

Abstract: The biological implications of bioinformatics can already be seen in various implementations. Biological taxonomy may seem like a simple science in which the biologists merely observe similarities among organisms and construct classifications according to those similarities^[1], but it is not so simple. By applying data mining techniques on gene sequence database we can cluster the data to find interesting similarities in the gene expression data. One of the applications of such kind of clustering is taxonomically clustering the organisms based on their gene sequential expressions. In this study we outlined a method for taxonomical clustering of species of the organisms based on the genetic profile using Principal Component Analysis and Self Organizing Neural Networks. We have implemented the idea using Matlab and tried to cluster the gene sequences taken from PAUP version of the ML5/ML6 database. The taxa used for some of the basidiomycetous fungi form the database. To study the scalability issues another large gene sequence database was used. The proposed method clustered the species of organisms correctly in almost all the cases. The obtained were more significant and promising. The proposed method clustered the species of organisms correctly in almost all the cases. The obtained results were more significant and promising.

Key words: Bioinformatics, taxonomy, gene sequence classification, data mining, data classification, clustering, principal component analysis

INTRODUCTION

Taxonomy: Biological taxonomy may seem like a simple science - Biologists merely observe similarities among organisms and construct classifications according to those similarities, but it is not so simple. Consider an obvious type of similarity referred to as 'morphological similarity': when organisms have a similar body shape and structure. Dogs have a different morphology than coyotes and dogs and coyotes are more similar to one another than either is to foxes. Mammals come in neat morphological packages; however, morphology is an inadequate marker for classifying many organisms, especially insects, molds, fungi and bacteria. For example, the fruit flies *Drosophila persimilis* and *Drosophila pseudoobscura* have nearly identical morphologies. It took years for biologists to determine that many organisms thought to be *Drosophila persimilis* are in fact members of a different species, *Drosophila pseudoobscura*. Matters get worse in bacteria. Some bacteriologists have thrown up their hands in classifying parasitic bacteria. The morphological differences between such bacteria grade into one another resulting in a continuum of organisms. Bacteria are not an exceptional case. Most of life on Earth, both in terms of biomass and biodiversity, is bacterial.

Perhaps a better foundation for biological classification can be found in genetics. We live in the

heady days of the Human Genome Project and other genome projects. Perhaps the organisms of one species are genetically more similar to one another than they are to organisms in other species. If this is true, then classification can be based on genetic similarity. There are, however, strong challenges to this suggestion; one being that genes are insufficient for distinguishing species. Turning to fruit flies again, there can be more genetic variation between different populations of a single fruit fly species than there is between two such species. In other words, two organisms in different species can be more similar to one another genetically than either is to the members of their own species.

Bioinformatics: Bioinformatics is a quickly growing field. It began out of necessity in the late 1960s and 1970s when scientists began sequencing genes and proteins. They soon realized that the amount of data would be too large for humans to interpret without the aid of computers. Databases were created to store the data and tools had to be developed to search them. Algorithms that could search this type of data were developed and implemented.

The biological implications of bioinformatics can already be seen in the simple existence and usage of the databases and search engines. These tools have sped up the scientific research. Now, biologists can compare their newly sequenced DNA with the DNA from many different species without months of research. The entire

human genome is available online to anyone who cares to search it. None of these things would have been possible without the development of the tools of bioinformatics.

The organization of information in bioinformatics is far from perfect; however, the Human Genome Project is a rare example of fairly well organized data. In most cases, scientists work on a few genes and then submit the sequences of those genes to the databases. Multiple copies of genes have been submitted and it may be difficult to tell what order they should be in and how they relate to each other.

DNA, RNA and proteins the bases of life and evolution can be traced through changes in their sequences. The search tools are needed to tell the related sequences from unrelated sequences. So substitution matrices are now used concurrently with the search tools. These matrices are based on the sets of related data and they help the search tools determine which sequences are most likely to be evolutionarily related.

Sequence analysis is the process, by which you use to find information about a nucleotide or amino acid sequence using computational methods^[2]. Common tasks in sequence analysis are the identification of genes; the determination of the similarity between two genes, the determination the protein coded by a gene and the determination the function of a gene by finding a similar gene in another organism with a known function. Determining the similarity between two sequences is a common task in computational biology. Starting with a nucleotide sequence for a human gene, this example uses alignment algorithms to locate a similar gene in another organism.

Classification and clustering: The term “classification” is frequently used as an algorithm for all data mining tasks^[1]. Instead, it is best to use the term to refer to the category of supervised learning algorithms used to search interesting data patterns. With classification algorithms have become very popular and ubiquitous in DM research, it is just but one of the many types of algorithms available to solve a specific type of DM task^[3,4].

Classification algorithms rely on human supervision to train itself to classify data into predefined categorical classes^[5]. For example, given classes of patients that corresponds to medical treatment responses; identify most responsive forms of treatment for the patient.

Principal components analysis: In this work, Principal Component Analysis is used for reducing the dimension of the data. The dimensionality reduction is also used for plotting the data in a two dimensional space. Reducing dimensionality combines variables that have a linear relationship, therefore reducing two variables to one. By combining two variables we reduce the ability

to investigate the interrelationship between all the variables.

PCA assumes that all the variables in a process should be used in the analysis. Therefore it becomes difficult to distinguish the important variable from the less important.

Principal Components

A data set $x_i, (i = 1, \dots, n)$ is summarized as a linear combination of orthonormal vectors (called principal components):

$$f(\mathbf{x}, \mathbf{V}) = \mathbf{u} + (\mathbf{x}\mathbf{V})\mathbf{V}^T$$

where $f(\mathbf{x}, \mathbf{V})$ is a vector valued function, \mathbf{u} is the mean of the data $\{x_i\}$ and \mathbf{V} is an $d \times m$ matrix with orthonormal columns. The mapping $z_i = x_i\mathbf{V}$ provides a low-dimensional projection of the vectors x_i if $m < d$.

As shown in Fig. 1 PCA estimates the projection matrix \mathbf{V} minimizing

$$R_{emp}(\mathbf{x}, \mathbf{V}) = \frac{1}{n} \sum_{i=1}^n \|\mathbf{x}_i - f(\mathbf{x}_i, \mathbf{V})\|^2$$

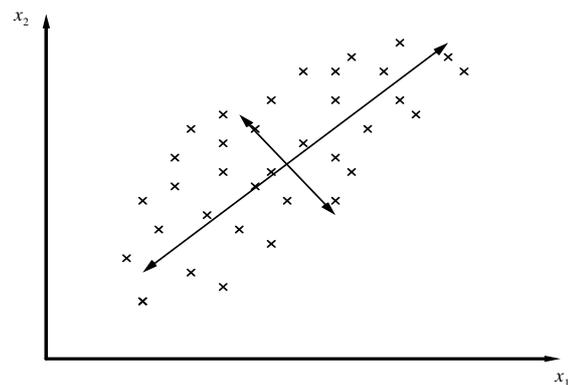


Fig. 1: The first principal component

The first principal component is an axis in the direction of maximum variance.

Properties of PCA: Principal components have the following optimal properties in the class of linear functions $f(x, V)$ ^[6].

- * The principal components Z provide a linear approximation that represents the maximum variance of the original data in a low-dimensional projection.
- * They also provide the best low-dimensional linear representation in the sense that the total sum of squared distances from data points to their projections in the space is minimized:
- * If the mapping functions F and G are restricted to the class of linear functions, the composition $F(G(x))$ provides the best (i.e., minimum empirical risk) approximation to the data.
- * PCA most appropriate for normal/elliptical distributions (where linear PCA approach provides the best possible solution).

Consequently, Principle Component Analysis (PCA) replaces the original variables of a data set with a smaller number of uncorrelated variables called the principle components. If the original data set of dimension D contains highly correlated variables, then there is an effective dimensionality, $d < D$, that explains most of the data. The presence of only a few components of d makes it easier to label each dimension with an intuitive meaning. Furthermore, it is more efficient to operate on fewer variables in subsequent analysis.

Using the built-in functions of Matlab we can do the PCA in simple steps or even in one step with new versions of Matlab.

Naïve k-means algorithm: One of the most popular heuristics for solving the k-means problem is based on a simple iterative scheme for finding a locally optimal solution^[7]. This algorithm is often called the k-means algorithm. There are a number of variants to this algorithm, so to clarify which version we are using, we will refer to it as the naïve k-means algorithm as it is much simpler compared to the other algorithms described here. This algorithm is also referred to as the Lloyd's algorithm.

The naïve k-means algorithm partitions the dataset into 'k' subsets such that all records, from now on referred to as points, in a given subset belonging to the same center. Also the points in a given subset are closer to that center than to any other center.

The partitioning of the space can be compared to that of Voronoi partitioning except that in Voronoi partitioning one partitions the space based on the distance and here we partition the points based on distance.

The algorithm keeps track of the centroids of the subsets and proceeds in simple iterations. The initial partitioning is randomly generated, that is, we randomly initialize the centroids to some points in the region of the space. In each iteration step, a new set of centroids is generated using the existing set of centroids following two very simple steps. Let us denote the set of centroids after the i^{th} iteration by $C(i)$. The following operations are performed in the steps:

- * Partition the points based on the centroids $C(i)$, that is, find the centroids to which each of the points in the dataset belongs. The points are partitioned based on the Euclidean distance from the centroids.
- * Set a new centroid $c(i+1) \in C(i+1)$ to be the mean of all the points that are closest to $c(i) \in C(i)$. The new location of the centroid in a particular partition is referred to as the new location of the old centroid.

THE PROPOSED METHOD

There are so many methods for data classification. Generally the selection of a particular method may

depend on the application. The selection of a particular methodology for data classification may depend on the volume of data and the number of classes present in that data. Further more, the classification algorithms are designed in a custom manner for a specific purpose to solve a particular classification scenario.

In addition, if the dimension of the data increases, then the problem becomes more complex and will take a very long time to get a meaningful result.

This work used the Principal Component Analysis for Feature Vector Selection from the Gene Sequence information. To cluster the Feature Vectors of the Gene Sequence Data, we used Self Organizing Feature Maps. The Fig. 2 explains the proposed method.

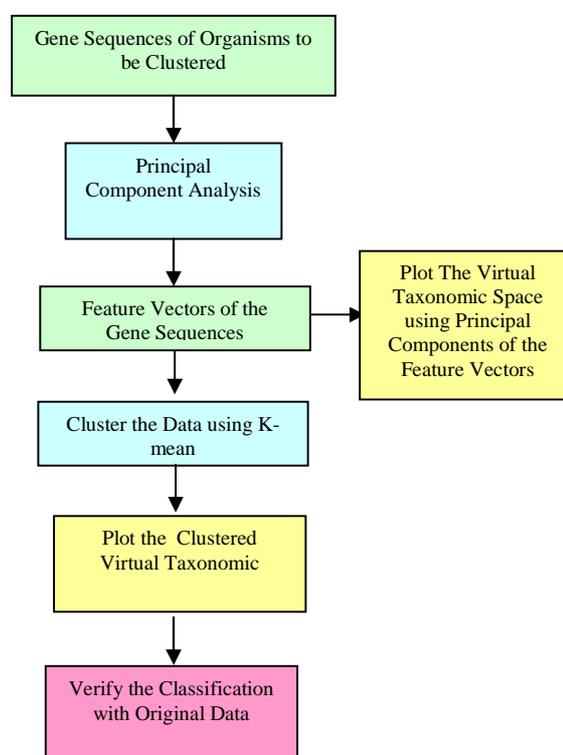


Fig. 2: The diagram explaining proposed method

EXPERIMENTAL RESULTS

About the gene sequence database: A biological database is a large, organized body of persistent data, usually associated with computerized software designed to update, query and retrieve components of the data stored within the system. A simple database might be a single file containing many records, each of which includes the same set of information. For example, a record associated with a nucleotide sequence database typically contains information such as contact name, the input sequence with a description of the type of molecule, the scientific name of the source organism from which it was isolated and often, the literature citations associated with the sequence.

There are many different database types, depending both on the nature of the information being stored and on the manner of data storage. In biological databases the data consist of literature information, nucleotide or protein sequences, structural information and so on. The types of data storage used are flat-files, relational databases or object-oriented databases (see lecture handouts).

To estimate the performance of the proposed clustering algorithm, gene sequences were used as shown in Table 1 and Fig. 3. The sequence was unaligned. So a uniform size of 150 character length of the sequence was used for dimensionality reduction. The following output shows one sample gene sequences in FASTA format.

```
>orf19.1162 Contig19-10097 (47732, 47355), reverse
complemented (378 nucleotides)
ATGGGAGAAGATCACGAATTTTACGGTGGCAT
CAAGTCATATGATGTGCACACATACTATGGTAA
TGAGAAGGAAGAAAAGTTGGCTTTTGCTTTAA
GGGAGAAAGTATTTAAAAGATTTTGCCAAAGAA
ATTGAGAATGGAGAGATAAGAGTTTACAAGTT
TTGGGAAAACCAATTGGTCCACATCCAATTA
GAATGTGGGAATTGGATTTCAAAGATCCTGAA
ATTTTCAAAGTTGTTGTTTCTTATTTCCAGTTGA
ACCATGGACCATTATCTGTTTTAATTCATCCAC
GGACCGACCAGGGTGATCTCAAGGATCACACG
GAACATGCCTTATGGTTAGGTCACAAAGTCCGT
TTGGATACCAGCTTGCTTTGA
```

Figure 4 shows the plotting of gene sequences in virtual taxonomic space.

Table 1: The average performance

| Number of Sequences | Time taken for dimensionality Reduction | Time taken for K-mean clustering | Total time for virtual taxonomical Space creation |
|---------------------|---|----------------------------------|---|
| 250 | 0.198 | 0.021 | 0.219 |
| 500 | 0.668 | 0.030 | 0.698 |
| 750 | 1.639 | 0.037 | 1.676 |
| 1000 | 2.624 | 0.037 | 2.661 |
| 1250 | 4.173 | 0.046 | 4.219 |

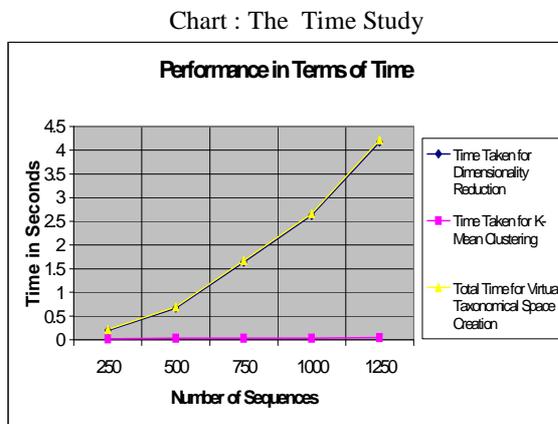
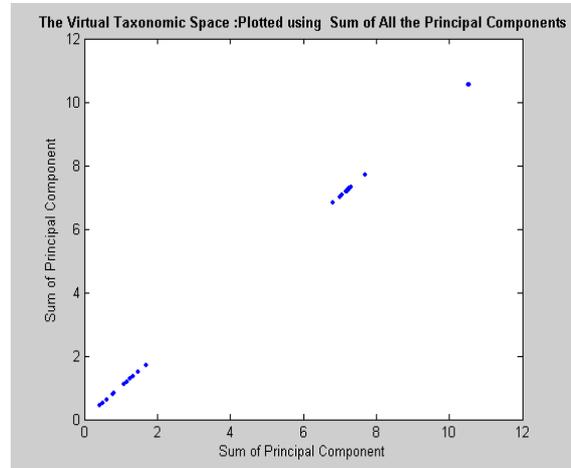
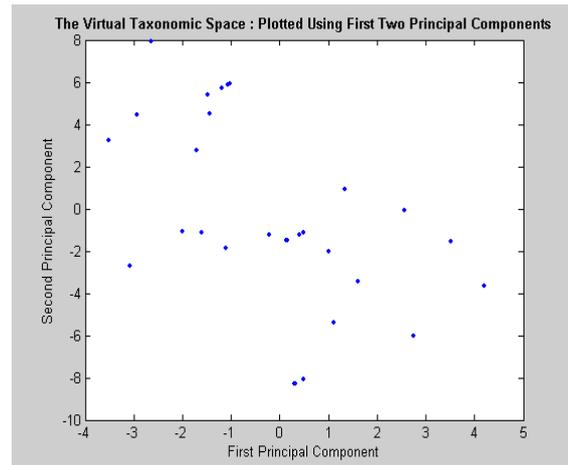


Fig. 3

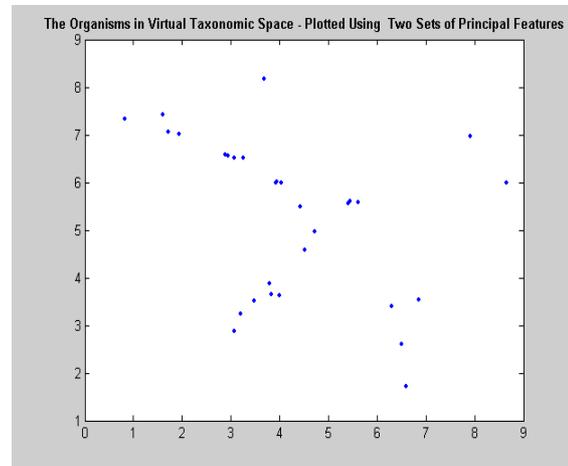
The virtual taxonomic space: Plotted using sum of all the principal components



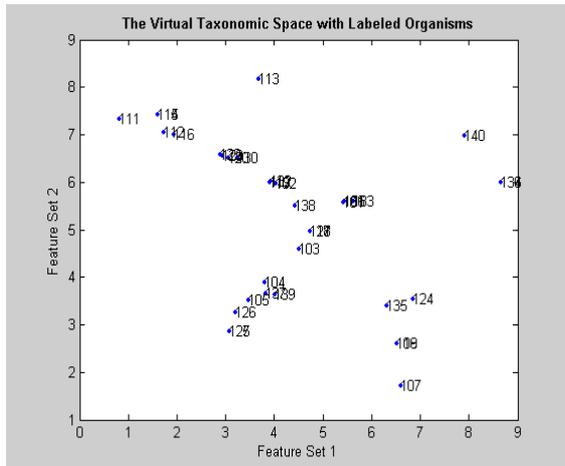
The virtual taxonomic space: Plotted using first two principal components



The organisms in virtual taxonomic space: Plotted using two sets of principal features



The virtual taxonomic space with labeled organisms



The organisms were clustered in virtual taxonomic space

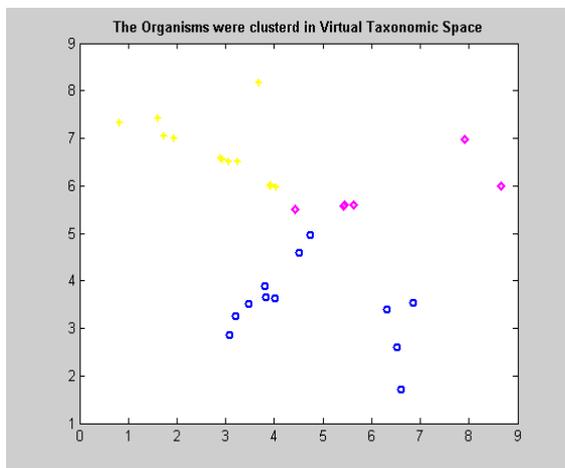


Fig. 4

CONCLUSION AND SCOPE FOR FURTHER ENHANCEMENTS

The proposed Gene Sequence Clustering algorithm has been implemented and tested successfully by using Matlab on Windows operating system as shown in Fig. 4. The Clustering Performance of the proposed algorithm was tested against huge Gene sequence databases gathered from Internet resources.

The main sequence database used to find the scalability of the proposed clustering algorithm was a Gene sequence database in Fasta format. As shown in the Fig. 3, the performance of the algorithm linearly changed with respect to the increase of number of sequences processed and there was very little time taken for k-means clustering while comparing the time taken for dimensionality reduction. So the total time for the completion of the whole algorithm is almost equal to the time required for dimensionality reduction. So the performance of the algorithm is very much related with the total number of sequences.

The issues related with very long sequence length and very huge database may be addressed in future work. In this research the scalability of the algorithm was tested. In future works the accuracy of classification of the algorithm can be verified with a suitable gene or protein sequence database.

REFERENCES

1. Guralnik, K., 2001. A scalable algorithm for clustering sequential data. ACM.
2. Guha, R., *et al.* 1998. CURE: An efficient clustering algorithm for large databases. DBLP.
3. Adriaans, D.Z.P., 2000. Data Mining. Addison Wesley.
4. Michel, J.A. Berry, 2003. Mastering Data Mining. Willey Eastern.
5. Zalzal, S. *et al.*, 2002. A genetic rule-based data clustering toolkit.
6. Teófilo Campos. PCA for face recognition. Creativision Research Group, IME- USP.
7. Vaidya, J. and C. Clifton, 2003. Proc. Ninth ACM SIGKDD Intl. Conf. on Knowledge discovery and data mining. Research track: Privacy-preserving k-means clustering over vertically partitioned data.