

# Method of Fuzzy Matching Feature Extraction and Clustering Genome Data

Nagamma Patil<sup>1+</sup>, Durga Toshniwal<sup>1</sup> and Kumkum Garg<sup>2</sup>

<sup>1</sup> Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India

<sup>2</sup> Department of Computer Science and Engineering, Manipal Institute of Technology, Manipal, India

**Abstract.** Cluster analysis divides data into groups that are meaningful and useful. Sequence clustering is needed and contributes to the assessment of gene and species evolutionary relationships. Clustering methods are thus necessary to carry out these identification operations in an accurate and fast way. In this paper, a method for feature extraction based on fuzzy matching is proposed and these features are used for genome data clustering. Given a database of genome sequences, our proposed work includes generating candidates of length equal to query, find total number of approximate matching patterns to query with given fault tolerance and then using this total number of matches for clustering. Fuzzy C-Means algorithm is used for genome data clustering. Genome data of two species namely Yeast and E. coli are used to verify proposed method.

**Keywords:** *n*-mer method, fuzzy matching, genome data, FCM.

## 1. Introduction

The main objective of the unsupervised learning technique is to find a natural grouping or meaningful partition using a distance function. Clustering is a technique which has been extensively applied in a variety of fields related to life science and biology. In sequence analysis, clustering is used to group homologous sequences into gene or protein families. There are many types of clustering techniques namely hierarchical clustering, partitional clustering, and fuzzy clustering [1].

Bioinformatics is a recent discipline which consists of analysing biological information. Identification of a species from its genomic database is a challenging task. Early approaches of comparing and grouping sequences are alignment methods. In fact, pair-wise alignment is used to compare and to cluster sequences. There are two types of pair-wise sequence alignments, local and global [2, 3]. Smith and Waterman local alignment algorithm [4] helps in finding conserved amino acid patterns in protein sequences. Needleman and Wunsch global alignment algorithm [5] attempts are made to align the entire sequence using as many characters as possible, up to both ends of each sequence. The pair-wise alignment is computationally expensive because of the large number of comparisons carried out. Therefore these methods are not efficient to cluster a large set of data [6]. In real world biological applications, most of the sequences are “similar” instead of exactly the same. Thus searching in sequence repositories often requires going beyond exact matching to determine the sequences which are similar [7, 8].

Recently we proposed a method for species identification based on approximate pattern matching [9]. The novelty in our previous work was feature extraction technique for genome data. In our method we extracted all candidate/subsequences that satisfy: length greater or equal to given minimum length, given number of mismatches and support greater or equal to user threshold. These frequent subsequences are used as features to construct a binary table where the presence or absence of an attribute/feature in a sequence is represented by 1 or 0 respectively. Classification of genome sequences has been done using data mining

---

<sup>+</sup> Corresponding author. Tel.: + 09997515988.  
E-mail address: nagamapatil@gmail.com.

techniques. Frequent approximate patterns are extracted from two different species namely the bacterium *Escherichia coli* (*E. coli*) [10], *Saccharomyces cerevisiae* (yeast) [11].

In this paper we propose a method for genome data clustering based on approximate matching. The paper explores a new approach to extract genomic features of a species from its genome sequence. Given a database of genome sequences, our proposed work includes generating candidates of length equal to query, find total number of approximate matching patterns to query with given fault tolerance for each sequence and then using this total number of matches for clustering. Fuzzy C-Means clustering algorithm is used for genome data classification. In this method we use a single vector as input to clustering. But in our previous method [9] we generated all candidates of length greater or equal to given minimum length, given support, given mismatch and then used all these candidates/features represented in binary form as explained above to classification. We apply our proposed method to genome data of *E.coli* and Yeast.

This paper is organized as follows: Related work is given in Section 2. In Section 3, proposed method is explained. Section 4 contains results and discussion along with data sets used. Finally, we draw conclusions in Section 5.

## 2. Related Work

There are two major approaches to biological sequence classification: alignment based and alignment free approach. Many alignment-based techniques have been developed, most notably are the Basic Local Alignment and Search Tool – BLAST, FASTA and position specific weight matrices. A comprehensive review of these techniques is given by Durbin [12]. While these techniques have proven a degree of success, they are limited by the high computational load when the number of sequences in the database against which the search is applied is large, and is also dependent on the length of individual sequences which may vary considerably between sequences.

Thus to compare long genomic sequences, linguistic-type methods, which are also often referred to as “free alignment methods”, have been developed. This approach was first described in [13]. The methods which dominate at present employ different versions of calculating the frequencies of fixed-length words in sequences being compared [14] and a later review [15]. A linguistic method, in any of its implementations, generates a pattern of word frequencies in the sequence. Since such pattern depends on the entire sequence, it can be viewed as being spread over the sequence. This pattern is usually used for the determination of the hierarchical system of the relationships between species.

In relative to alignment free techniques several methods are presented in literature to identify different organisms based on their genome data. A set of 16 kinds of dinucleotide compositions was used to analyze the protein-encoding nucleotide sequences in nine complete genomes. The dinucleotide composition was significantly different between the organisms. The distribution of genes from an organism was clustered around its center in the dinucleotide composition space [16].

Classification of organisms into 2 classes- Bacteria and Archea, based on their di-nucleotide frequencies in DNA sequence using Naive Bayesian approach was discussed by Sandberg et al. [17]. The methodology is based on scanning all genomes for the occurrences of all possible overlapping motifs with a length of  $n$  nucleotides. Then, a genomic sequence is chosen at random from anywhere inside a genome. From this genomic sequence, all overlapping motifs are extracted. The Naive Bayesian classifier uses the extracted motifs to predict their most probable genomic origin by comparing the frequencies of the extracted motifs with the motif frequencies of the different genomes.

Narasimhan et al. [18] designed a scheme for automatic identification of a species from its genome sequence. A set of 64 3-tuple keywords is first generated using the 4 types of bases A, T, C and G. These keywords are searched on randomly sampled genome sequence of a given length (10,000 elements) and frequency count for each of the keywords is determined to obtain a DNA-descriptor. Principal Component Analysis was employed on the frequency counts for  $N$  sampled instances to obtain a unique feature descriptor which identifies the species from its genome sequence. It was shown that the feature descriptors were effective representatives of the structural signature of the species. The problem with all above  $n$ -gram based classification is number of features to classifier increases as pattern length  $n$  increases.

### 3. Proposed Method

Given a sequence database with  $N$  number of sequences, query pattern of length  $p$  and percentage of fault tolerance allowed  $k$ . We want to find, for every sequence in the database, total number of candidate patterns that will approximately match to the query pattern within tolerance (at most  $k$ ). First, candidates (subsequence) of length equal to query are generated. For a given sequence, the set of the candidates with length  $n$  can be generated by sliding a window of  $n$  characters on the whole sequence. This movement is carried out character by character. With each movement a subsequence of  $n$  characters is extracted.

Once candidates are generated, we are going to match each candidate to the query to find Levenshtein distance between them (let  $s$  be the candidate pattern and  $t$  be the query pattern, the Levenshtein distance is the number of deletions, insertions, or substitutions required to transform  $s$  into  $t$ ). If the distance is less than or equals the specified tolerance, then candidate pattern is counted as fuzzy/approximate match. This process is repeated for entire database. Select query pattern that is frequent in given database of sequences. Once all the approximate matching patterns are extracted, we are going to build a feature table where each row corresponds to sequence the value in column for a particular sequence corresponds to number of fuzzy matches. In the mining step, FCM algorithm is applied to the learning context to identify genome data into different groups.

Example: Consider the following sequence

Seq<sub>1</sub>: ATTCGGTACA

Let the query be ATTC of length 4 and tolerance allowed is 50% (maximum distance allowed is 2). To encode the given sequences we have to first generate candidates of length equal to query and then we have to find Levenshtein distance between each candidate and query as shown in Table1. Then finally count total fuzzy matches within given tolerance. Fuzzy matches within the given tolerance are marked as bold in Table1. Total fuzzy matches for candidates generated with length 4(L4) are 3. Table 2 shows final feature table for the example sequence and this single vector feature table is used as input to FCM. This type of pre-processing method is used in [19] for recognition of genome data by Naïve Bays classifier but in this paper these extracted features are used as input to Fuzzy C-Means algorithm.

Table. 1: Candidates generated and their respective distance with query ATTC for the example Seq1.

| Candidates Generated with Length 4 | Distance |
|------------------------------------|----------|
| ATTC                               | <b>0</b> |
| TTCG                               | <b>2</b> |
| TCGG                               | 4        |
| CGGT                               | 4        |
| GGTA                               | 3        |
| GTAC                               | <b>2</b> |
| TACA                               | 3        |

Table. 2: Feature table constructed from example sequence (Input to FCM)

| Sequence Number  | Total fuzzy matches with Length 4 |
|------------------|-----------------------------------|
| Seq <sub>1</sub> | 3                                 |

#### 3.1. Algorithm used for clustering

One of the most widely used fuzzy clustering algorithms is the Fuzzy C-Means (FCM) algorithm. This method was developed by Dunn [20] and improved by Bezdek [21] and is frequently used in pattern recognition. FCM is an iterative algorithm. The aim of FCM is to find cluster centers (centroids) that minimize a dissimilarity function. To initiate the fuzzy partitioning, the membership matrix ( $U$ ) is randomly initialized according to (1).

$$\sum_{i=1}^c u_{ij} = 1, \forall j = 1, \dots, n \quad (1)$$

The algorithm minimizes a dissimilarity (or distance) function which is given in (2).

$$J(U, c_1, c_2, \dots, c_c) = \sum_{i=1}^c j_i = \sum_{i=1}^c \sum_{j=1}^n u_{ij}^m d_{ij}^2 \quad (2)$$

where  $u_{ij}$  is between 0 and 1,  $c_i$  is the centroid of cluster  $i$ ,  $d_{ij}$  is the Euclidian distance between  $i^{th}$  centroid and  $j^{th}$  data point and  $m$  is a weighting exponent.

To reach a minimum of dissimilarity function there are two conditions. These are given in (3) and (4)

$$c_i = \frac{\sum_{j=1}^n u_{ij} x_j}{\sum_{j=1}^n u_{ij}^m} \quad (3)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left( \frac{d_{ij}}{d_{kj}} \right)^{2/(m-1)}} \quad (4)$$

By iteratively updating the cluster centers and the membership grades for each data point, FCM iteratively moves the cluster centers to the optimal location within a data set.

#### 4. Results and Discussion

In our experiment, we used complete genome data of two species namely, bacterium *Escherichia coli* (E. coli) and the *Saccharomyces cerevisiae* (Yeast). Since the complete genome data of species is very long, it is sampled into sizes of 10,000 bp. The proposed approach is tested with 200 samples. (100 samples are from E.coli and 100 samples are from Yeast). To extract approximate matching patterns from genome sequences, we used a query of length 10 and allowed tolerance from 10% to 70%. The proposed approach is tested separately for approximate matches extracted with each value of the fault tolerance and the accuracies (correctly clustered genome samples \*100/ total number of genome samples) are monitored. In each case, the experimental results vary according to the tolerance allowed. Clustering parameters used in our experiment are, exponent for the partition matrix U: 2.0, maximum number of iterations: 100, minimum amount of improvement: 1e-5 and number of clusters: 2.

Table 3 shows experimental results for 200 samples with fault tolerance from 10% to 70%. It can be observed from the obtained results that, accuracy of correctly classified genome samples increases with increase in fault tolerance. Our results show that, the highest accuracy achieved is 98% with sampling/sequence size of 10,000bp and with allowed tolerance of 70%. Fig. 1 shows, graphical representation of the percentage accuracy obtained at different tolerances. It can be observed that as tolerance increases, number of correctly classified data points increases.

Table. 3: Experimental results for different tolerances

| PT | No. of ICS | No. of CCS | PA   | PM   |
|----|------------|------------|------|------|
| 10 | 88         | 112        | 56   | 44   |
| 20 | 69         | 131        | 65.5 | 34.5 |
| 30 | 31         | 169        | 84.5 | 15.5 |
| 40 | 9          | 191        | 95.5 | 4.5  |
| 50 | 5          | 195        | 97.5 | 2.5  |
| 60 | 5          | 195        | 97.5 | 2.5  |
| 70 | 4          | 196        | 98   | 2    |

PT:Percentage Tolerance; ICS: Incorrectly Classified Samples; CCS: Correctly Classified Samples; PA:Percentage Accuracy; PM:Percentage Misclassification

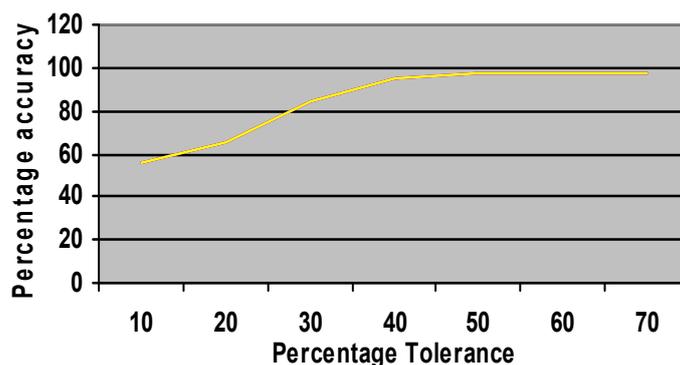


Fig. 1: Performance of FCM at different tolerance allowed

When the allowed tolerance is only 10%, since our query pattern is of length 10, we generated the candidates of length 10 from genome samples. 9 characters in the candidate must match to query pattern for the given tolerance of 10%. In this case, the percentage of matching of candidate to query is very less. Hence FCM resulted in less accuracy.

A tolerance of 20% indicates that the allowed distance in matching a candidate to a query is 2. It indicates that when candidates generated are of length 10, minimum of any 8 characters must match the query pattern. Hence, for this case, fuzzy match between the candidates generated and query will increase compared to 10% tolerance. Therefore there is increase in classification accuracy compared to 10% tolerance. Similarly when the allowed tolerance is 30%, 40%, 50%, 60% and 70% maximum distance allowed in matching candidate to query is 3, 4, 5, 6 and 7 respectively.

At 70% tolerance the allowed distance 7 indicates that minimum of 3 characters must match to query. The mismatch allowed in matching a candidate to query is increased, hence number of fuzzy matches will increase. In this case, our experimental results show that the maximum classification accuracy 98% by FCM.

## 5. Conclusions

Bioinformatics is a recent discipline which consists of analysing biological information. Identification of a species from its genomic database is a challenging task. In our proposed method a single vector (value representing number of approximate matches to query for given sequence) is used as input to FCM algorithm. To extract approximate matching patterns from genome sequences, we have used query of length 10 and varied tolerance. Our experimental results show that, the highest accuracy achieved is 98% for sampling/sequence size of 10,000bp and with allowed tolerance of 70%.

## 6. References

- [1] J. Han, and M. Kamber. Data mining: concepts and techniques. 2nd edition. Morgan Kaufmann Publishers, San Francisco, 2006.
- [2] P. Clote, and R. Backofen. Computational Molecular Biology – An Introduction. John Wiley & Sons, Ltd, 2000.
- [3] D. W. Mount. Bioinformatics – Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, New York, 2002.
- [4] T. F. Smith, and M. S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 1981, 147:195-197.
- [5] S. B. Needleman, and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of the proteins. *J. Mol. Biol.*, 1970, 48:443-453.
- [6] S. Fayeche, N. Essoussi, and M. Limam. Partitioning clustering algorithms for protein sequence data sets. *BioData Mining*, 2009, 2:3.
- [7] G. Navarro. A guided tour to approximate string matching. *ACM Computing Surveys*, 2001, 33: 31-88.
- [8] M. Hassan, and A. Muneer. Genome Sequence Analysis: A Survey. *J. of Computer Science*, 2009, 5 (9): 651-660,

Science Publications.

- [9] N. Patil, D. Toshniwal, G. Kumkum. Species Identification Based on Approximate Matching. 4th Annual ACM Bangalore Conf., Compute 2011, March 25-26, Bangalore, India. .
- [10] F. R. Blattner, G. Plunkett, C. A. Bloch, N. T. Perna, V. Burland, and M. Riley. The complete genome sequence of *Escherichia coli*. *Science*, K-12, 277:1453-1462.
- [11] <http://downloads.yeastgenome.org>
- [12] R. Durbin, S. R. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis*. Cambridge University Press, 1998.
- [13] V. Brendel, J. S. Beckmann, and E. N. Trifonov. Linguistics of nucleotide sequences: morphology and comparison of vocabularies. *J. Biomol. Struct. Dyn.*, 1986, 4:11–21.
- [14] T. Wu, J. Burke, and D. Davison. A measure of DNA sequence dissimilarity based on mahalanobis distance between frequencies of words, *Biometrics*, 1997, 53:1431–1439.
- [15] S. Vinga, and J. Almeida. Alignment-free sequence comparison—a review. *Bioinformatics*, 2003, 19(4):513–523.
- [16] H. Nakashima, M. Ota, K. Nishikawa, and T. Ooi. Genes from nine genomes are separated into their organisms in the dinucleotide composition space. *DNA Res.*, 1998, 5:251–259.
- [17] R. Sandberg, G. Winberg, C. I. Branden, A. Kaske, I. Ernberg, and Coster. Capturing Whole - Genome characteristics in short sequences using a naive Bayesian classifier. *Genome Res.*, 2001, 11:1404-09.
- [18] S. Narasimhan, S. Sen, and Konar. Species identification based on mitochondrial genomes. In *Proceedings of International Conference of Cognition and Recognition*. Mysore, India, December 22- 23, 2005.
- [19] N. Patil, D. Toshniwal, G. Kumkum. Recognition of an organism from its genome data based on fuzzy matching features. *International Conference on Information and Communication Technology*, Chennai, India, December 24, 2011.
- [20] J. C. Dunn. A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters. *J. of Cybernetics*, 1973, 3:32-57.
- [21] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*, Plenum Press, NY, 1981.