A Novel Scalable Feature Extraction Approach for COVID-19 Protein Sequences and their Cluster Analysis with Kernelized Fuzzy Algorithm

1st Preeti Jha
Computer Science and Engineering
Indian Institute of Technology Indore
Indore, India
jha.preeti07@gmail.com

2nd Aruna Tiwari
Computer Science and Engineering
Indian Institute of Technology Indore
Indore, India
artiwari@iiti.ac.in

3rd Neha Bharill
Computer Science and Engineering
Mahindra University
Hyderabad, India
neha.bharill@mahindrauniversity.edu.in

4th Milind Ratnaparkhe
Biotechnology
ICAR-IISR
Indore, India
milind.ratnaparkhe@gmail.com

5th Om Prakash Patel
Computer Science and Engineering
Mahindra University
Hyderabad, India
omprakash.patel@mahindrauniversity.edu.in

6th Nilagiri Harshith
Computer Science and Engineering
Mahindra University
Hyderabad, India
harshith170530@mechyd.ac.in

7th Soundarya Lahari Solasa
Electrical and Electronics
Mahindra University
Hyderabad, India
soundarya170250@mechyd.ac.in

2022 IEEE International Conference on Big Data and Smart Computing (BigComp) | 978-1-6654-2197-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/BIGCOMP54360.2022.00021

***Abstract*—COVID-19 (Coronavirus Disease-19), a disease caused by the SARS-CoV-2 virus, was declared a pandemic by the World Health Organization on March 11, 2020. To solve the global problem of analysis of different variants of COVID-19 genome sequences, there is a need to develop intelligent, scalable machine learning techniques that can process and analyze important COVID-19 protein data by utilizing the Big Data framework. For this, we have first proposed a feature extraction approach for COVID-19 protein data named Scalable Distributed Co-occurrence-based Probability-Specific Feature extraction approach (SDCPSF). The proposed SDCPSF approach is executed on the Apache Spark cluster to preprocess the massive COVID-19 protein sequences. The proposed SDCPSF represents each variable-length COVID-19 protein sequence with fixed length six dimensions numeric feature vectors. Then the extracted features are used as input to the kernelized fuzzy clustering algorithms, i.e., KSRSIO-FCM and KSLFCM, which efficiently performs clustering of big data due to its in-memory cluster computing technique and thus forms clusters of COVID-19 genome sequences. Furthermore, the performance of KSRSIO-FCM is compared with another scalable clustering algorithm, i.e., KSLFCM, in terms of the Silhouette index (SI) and Davies-Bouldin index (DBI).**

***Keywords*-Feature Extraction; COVID-19 Protein Sequences; Apache Spark Cluster; Kernelized Fuzzy Clustering**

I. Introduction

The Coronavirus Disease 2019 (COVID-19) pandemic has placed immense stress on the world's healthcare system. The COVID-19 data in terms of protein sequences are growing faster than the rate at which it can be analyzed. Protein sequences contain characters from the 20-letter amino acid alphabets ∑={A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y}. It is becoming increasingly popular to mine valuable information from huge COVID-19 genomics data to interpret data in a useful and timely manner. To mine useful information from huge genomics data such as protein sequences many machine learning (ML) approaches like clustering, classification, and neural network are widely applied [1]. Before applying any machine learning algorithm on COVID-19 protein sequences the encoding of protein sequences in terms of feature vectors is an important issue. The high dimensionality of protein data during the implementation of machine learning algorithms tends to create many important problems for researchers [2]. A good input representation (extraction of features) is necessary for the proper classification of protein sequences. G Wang [3] proposed a technique for feature extraction, which tries to capture the global similarity that refers to the overall similarity among multiple sequences and the local similarity which refers to frequently occurring sub-strings in the sequences. Bharill [4] developed an approach to extract a six-dimensional numerical feature vector from a protein sequence. Many feature selection techniques have been introduced in the past, but, none of them is scalable. To handle COVID-19 protein data of high dimension, there is a need to design a scalable feature extraction approach that can represent each high dimensional genome sequence with a fixed dimensional numeric feature vector that can statistically select the significant features from a huge protein sequence. To design a scalable feature extraction approach, there is a need of integrating Big Data

DOI 10.1109/BigComp54360.2022.00021

processing framework in the feature extraction approach. Apache Spark is one such distributed framework for Big Data Processing [5]. It keeps the advantages of MapReduce scalable, which makes it more adaptable, and quicker, and simpler to use than other frameworks.

To design the scalable feature extraction approach integrated with big data processing framework, we have extended the work of Bharill [4] to propose a Scalable Distributed Co-occurrence based Probability-Specific Feature extraction approach (SDCPSF) for COVID-19 protein sequences. The advantage of the proposed method is that it represents each variable length protein sequence of huge size with a fixed-length numeric vector. In addition to this, it considers all possible position-specific variations of amino acids in COVID-19 protein sequences. Furthermore, the preprocessed COVID-19 protein sequences are used for cluster analysis. To perform the clustering of protein sequences for cluster analysis several clustering algorithms have been applied by researchers on COVID-19 and genome data [1, 6, 7]. In this work, we utilized the KSRSIO-FCM [8] method to cluster COVID-19 protein sequences in this study because KSRSIO-FCM works well on both linearly and non-linearly separable data and thus creates high-quality clusters. Also, the KSRSIO-FCM is a scalable clustering algorithm that integrates Apache Spark Big Data processing framework to process huge COVID-19 protein data.

This paper is organized as follows: Section II covers protein sequence feature extraction technique. The discussion of the proposed SDCPSF extraction algorithm on Apache Spark is discussed in section III. The experimental results are reported in terms of SI and DBI indexes in section IV. Finally, the conclusions of our work are drawn in section V.

II. Related Work

In this section, we are presenting a discussion of feature extraction method for protein sequences; Co-occurrence based Probability-Specific Feature (CPSF) [4] approach. The CPSF approach extracts features from the protein dataset in three steps as follows: In the first step of the CPSF approach [4], each protein sequence is encoded and presented in terms of six exchange groups. Exchange groups are efficient amino acid equivalent classes, formally represented by $\{e_1, e_2, e_3, e_4, e_5, e_6\}$, where e_1=$\{H, R, K\}$, e_2=$\{D, E, N, Q\}$, e_3=$\{C\}$, e_4=$\{S, T, P, A, G\}$, e_5=$\{M, I, L, V\}$ and e_6=$\{F, Y, W\}$ [3]. In the second step, we calculate the global similarity measure by calculating the probability of exchange groups at each position to the total number of protein sequences. The global similarity measures are calculated as follows:

$$(Probability)_{ij} = (Instance)_{ij}/\eta \tag{1}$$

Where $(Probability)_{ij}$ denotes the probability of instance of the i^{th} exchange group at j^{th} position, $(instance)_{ij}$ represents the frequency at which the i^{th} exchange group appears at j^{th} position and η represents the total number of sequences in a particular species. In the third stage of the CPSF approach [4], the local similarity measure is calculated, which determines each exchange groups location-specific weight within the sequence considering the weight factors. The weight of each exchange group can be calculated using the below formula:

$$(Weight)_i^{SEQ_k} = \sum_{j=1}^{j'} (Probability)_{ij} \times (PW)_{ij}^{SEQ_k} \tag{2}$$

Where $(Weight)_i^{SEQ_k}$ represents the weight of i^{th} exchange group corresponding to the k^{th} protein sequence, $(Probability)_{ij}$ denotes the probability of occurrence of the i^{th} exchange group at j^{th} position and $(PW)_{ij}^{SEQ_k}$ is the positional weight assigned to the i^{th} exchange group based on the presence of k^{th} protein sequence at j^{th} position. The CPSF approach represents the protein sequence with a feature vector consisting of only six numeric features. The scalable distributed version of the CPSF extraction method is presented next.

III. Proposed Work

This section describes the scalable algorithm implemented on Apache Spark cluster. To propose a scalable protein preprocessing algorithm for COVID-19 data, we followed the CPSF approach discussed by Bharill [4] and applied this approach to COVID-19 protein data to extract six numeric features. To make the protein preprocessing algorithm a scalable algorithm, we executed it on Apache Spark cluster and termed it as a scalable protein preprocessing algorithm: SDCPSF. The output obtained from the SDCPSF preprocessing algorithm is used as an input to the KSRSIO-FCM/KSLFCM [8] clustering algorithm for forming clusters from COVID-19 protein sequences. For this, initially, the KSRSIO-FCM algorithm partitioned the entire COVID-19 protein dataset into various subsets and then clustering is performed on each subset by considering the similarity among the COVID-19 protein sequences. KSLFCM, on the other hand, clusters all of the data at once. Hence, KSRSIO-FCM has a lesser run-time than KSLFCM since it clusters a smaller portion of data in each subset rather than the entire data. The SDCPSF; a feature extraction algorithm for preprocessing of the COVID-19 dataset is explained as follows: The input given is a raw protein dataset. The output is a feature vector of the given input dataset, a file containing six numeric features. The SDCPSF extraction approach is explained in Algorithm 1. Line 1 of Algorithm 1, distributes the $COVID19Protein.txt$ dataset on Apache Spark clusters. The encoding of protein amino acid is performed using the stage one technique from Line 2-4, as explained in section II. In Line 5, the global similarity matrix is calculated. The probability matrix of the sequences on a master machine is then obtained without distributing the dataset in spark clusters. Again, distributes the result obtained from the second stage using Apache Spark clusters to find local similarity matrix. The global similarity matrix and local similarity calculation is explained in section as explained in II. Obtain the value of the exchange group from the index value and store it as a column number in Line 7. Then modify the key value of the exchange group by adding the previous value with the value of the probability dataframe at the position of the exchange group and column number in Line 8. The weight is computed using the local similarity matrix in Line 9. Finally, a vector consist of 6-dimensional numerical features are saved in a file using Line 10. The overall time complexity of proposed SDCPSF is O(n^2).

Algorithm 1 : *SDCPSF Algorithm*

```
Input: COVID-19 protein data: $COVID19Protein.txt$
Output: preprocessed protein data: $Feature\_Vectors.txt$
1: Distribute COVID-19 protein data in worker nodes.
2: Read the sequences from the file as Numpy arrays and parallelize using Apache Spark.
3: Store the data in the dataframe and split the each letter in the sequence to different columns.
3: Create an empty Probability dataframe with index names $e_1, e_2, e_3, e_4, e_5, e_6$.
4: Replace the particular amino acids with index names.
5: Calculate global similarity matrix using Eq.1.
6: Distribute the data obtained from global similarity matrix.
7: Get the value of exchange group from index value and store it as column number.
8: Modify the key value of the exchange group.
9: The weight of each exchange group is calculated using Eq. 2.
10: Save feature vectors in file $Feature\_Vectors.txt$.
```

The proposed SDCPSF has the significant characteristics that it takes raw protein sequences as input and produces 6-dimensional numeric feature vectors output in much less time using Apache Spark framework. The proposed SDCPSF method computes both local and global similarity for all potential position-specific changes of amino acids in COVID-19 protein sequences using Apache Spark cluster. In section IV, we present the experimental results applied to various protein datasets. The KSRSIO-FCM/KSLFCM algorithm takes the preprocessed 6-dimensional numeric feature vectors of huge COVID-19 protein sequences as input and produces output in terms of clusters.

IV. Experimental results

In the experiments, we analyzed the performance in terms of SI and DBI indexes of the proposed SDCPSF extraction method applied to the KSRSIO-FCM and KSLFCM algorithms on Apache Spark clusters. The assessment is carried out using an Apache Spark cluster. One master and five worker nodes comprise the Apache spark cluster.

A. Dataset description

In the experimental study, the two COVID-19 protein datasets (MERS and UniProt COVID-19) were used and preprocessed using the proposed SDCPSF method and then applied to the KSRSIO-FCM and KSLFCM to perform the clustering of COVID-19 datasets. The MERS dataset is Middle East respiratory syndrome-related coronavirus obtained from NCBI[1]. The number of sequences in the MERS dataset is 2850, and the size is 3952 KB. The UniProt COVID-19 protein dataset is downloaded from the COVID-19 data portal[2]. The number of sequences in the Uniprot COVID-19 dataset is 90, and the size is 71 KB.

B. Performance evaluation

The Silhouette Index (SI) and Davies-Bouldin Index (DBI) are used for evaluating the performance of clustering [9]. Both metrics are important for validating consistency among genomic data clusters. SI is a metric that compares how similar a data sample is to its cluster to other clusters. The Silhouette value is limited to a number between -1 and 1. A negative number implies poor clustering quality, whereas a positive value suggests excellent clustering quality. DBI divides a single record into two measures, one for the dispersion of individual clusters and the other for the partitioning of distinct clusters. Because the DBI is not constrained inside a particular range, a lower DBI implies higher clustering quality.

Table I: SI and DBI of MERS COVID-19 protein dataset applied to KSRSIO-FCM and KSLFCM algorithm.

#Cluster	**SI**		**DBI**	
	KSRSIO-FCM	**KSLFCM**	**KSRSIO-FCM**	**KSLFCM**
2	0.9226	0.7278	0.2847	0.6114
3	0.9596	0.5646	0.0369	0.614
4	0.9263	0.7681	0.0755	0.2781
5	0.8182	0.8022	0.3958	0.2967
6	0.8113	0.7953	0.3006	0.7024
7	0.8559	0.8531	0.1467	0.5848
8	0.8155	0.8127	0.2475	0.6096
9	0.8002	0.7651	0.3112	0.7209
10	0.8231	0.76839	0.4896	0.9755

C. Results and discussion

In this section, we compute the effectiveness of SDCPSF extraction method applied to the KSRSIO-FCM and KSLFCM algorithms on COVID-19 protein data in terms of SI and DBI indexes. We perform clustering of KSRSIO-FCM with subsets 3, where the subsets are the chunks of the entire data. And KSLFCM performs clustering of the whole

[1] https://www.ncbi.nlm.nih.gov/datasets/coronavirus/genomes/

[2] www.covid19dataportal.org

Table II: SI and DBI of UniProt COVID-19 protein dataset applied to KSRSIO-FCM and KSLFCM algorithm.

#Cluster	**SI**		**DBI**	
	KSRSIO-FCM	**KSLFCM**	**KSRSIO-FCM**	**KSLFCM**
2	0.3814	0.356	2.1967	2.4268
3	0.4936	0.4936	0.9827	0.9827
4	0.5042	0.4333	0.6679	0.9823
5	0.3813	0.3778	0.9721	0.9725
6	0.4349	0.4016	0.7189	0.9772
7	0.371	0.371	0.9671	0.9671
8	0.3407	0.3699	0.9613	0.9698
9	0.3141	0.372	0.958	0.9732
10	0.2965	0.3611	0.9885	0.9753

data. The clustering is performed on clusters ranging from 2 to 10. Table I tabulates the MERS COVID-19 protein dataset result in terms of SI and DBI applied to KSRSIO-FCM in comparison with KSLFCM. Observing the SI values, the KSRSIO-FCM algorithm has obtained better values than KSLFCM. On the other hand, the KSRSIO-FCM achieved a higher value for cluster 3 on the MERS dataset. In Table I, we have also reported the results of the MERS COVID-19 dataset on the protein dataset in terms of DBI. The value achieved by KSRSIO-FCM is lower than KSLFCM for almost all the clusters. In this way, we can conclude that the proposed SDCPSF performs better when applied to the KSRSIO-FCM algorithm in terms of SI and DBI values for MERS COVID-19 protein dataset. Table II tabulates results in terms of SI and DBI of the Uniprot COVID-19 protein dataset applied to KSRSIO-FCM in comparison with KSLFCM. Observing the SI values, the KSRSIO-FCM algorithm has obtained better values than KSLFCM. On the other hand, the KSRSIO-FCM achieved a higher value for cluster4 on the UniProt COVID-19 protein dataset. In Table II, we have also reported the results of Uniprot COVID-19 protein dataset in terms of DBI. The value achieved by KSRSIO-FCM is lower than KSLFCM for almost all the clusters. In this way, we can conclude that the proposed SDCPSF performs better when applied to the KSRSIO-FCM algorithm in terms of SI and DBI values for the Uniprot COVID-19 protein dataset.

V. Conclusion

In this paper, Apache Spark-based scalable feature extraction technique (SDCPSF) has been proposed to extract numerical feature vectors from massive COVID-19 protein sequences. After that, preprocessed numerical feature vectors are applied to the fuzzy clustering technique. In this case, we have used the KSRSIO-FCM and KSLFCM algorithms to cluster COVID-19 protein sequences. One distinctive characteristic of the proposed SDCPSF approach is that it computes both local and global similarity to take into account all possible position-specific variations of amino acids in a COVID-19 protein sequence using Apache Spark. Another essential characteristic is representing each variable-length protein sequence consisting of a long chain of amino acids with a fixed-length numeric vector of only six dimensions. We directed the exact evaluation of the SDCPSF algorithm applied to KSRSIO-FCM and compared it with KSLFCM on COVID-19 protein datasets, which exhibited potential advantages for utilizing our methodology for clustering protein sequences. In the future, the scalable feature extraction approach can be applied for handling massive protein sequences in the range of terabytes and petabytes for the clustering of COVID-19 datasets.

References

[1] M. R. Mahmoudi, D. Baleanu, Z. Mansor, B. A. Tuan, and K.-H. Pho, "Fuzzy clustering method to compare the spread rate of covid-19 in the high risks countries," *Chaos, Solitons & Fractals*, vol. 140, p. 110230, 2020.

[2] M. Steinbach, L. Ertöz, and V. Kumar, "The challenges of clustering high dimensional data," in *New directions in statistical physics*. Springer, 2004, pp. 273–309.

[3] J. T. L. Wang, Q. Ma, D. Shasha, and C. H. Wu, "New techniques for extracting features from protein sequences," *IBM Systems Journal*, vol. 40, no. 2, pp. 426–441, 2001.

[4] N. Bharill, A. Tiwari, and A. Rawat, "A novel technique of feature extraction with dual similarity measures for protein sequence classification," *Procedia Computer Science*, vol. 48, pp. 795–801, 2015.

[5] S. Tang, B. He, C. Yu, Y. Li, and K. Li, "A survey on spark ecosystem for big data processing," *arXiv preprint arXiv:1811.08834*, 2018.

[6] S. F. Ardabili, A. Mosavi, P. Ghamisi, F. Ferdinand, A. R. Varkonyi-Koczy, U. Reuter, T. Rabczuk, and P. M. Atkinson, "Covid-19 outbreak prediction with machine learning," *Algorithms*, vol. 13, no. 10, p. 249, 2020.

[7] A. Albahri, R. A. Hamid, J. K. Alwan, Z. Al-Qays, A. Zaidan, B. Zaidan, A. Albahri, A. AlAmoodi, J. M. Khlaf, E. Almahdi *et al.*, "Role of biological data mining and machine learning techniques in detecting and diagnosing the novel coronavirus (covid-19): a systematic review," *Journal of medical systems*, vol. 44, pp. 1–11, 2020.

[8] P. Jha, A. Tiwari, N. Bharill, M. Ratnaparkhe, M. Mounika, and N. Nagendra, "A novel scalable kernelized fuzzy clustering algorithms based on in-memory computation for handling big data," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020.

[9] N. Bolshakova and F. Azuaje, "Cluster validation techniques for genome expression data," *Signal processing*, vol. 83, no. 4, pp. 825–833, 2003.