The 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications
22-25 September, 2021, Cracow, Poland

Markov Clustering Algorithms and Their Application in Analysis of PPI Network of Malaria Genes

Mamata Das, PJA Alphonse, Selvakumar Kamalanathan
NIT Trichy, India, dasmamata.india@gmail.com {alphonse, kselvakumar}@nitt.edu

***Abstract* — An empirical study on three graph-based clustering algorithms has been presented here. We have discussed those three clustering algorithms and made an analysis. The algorithms we have considered here are: Markov Clustering (MCL), Regularized Markov Clustering Algorithm (R-MCL), and Variable Inflation MCL (VI-MCL). We have used two types of graph networks: random network generated using synthetic data and PPI networks generated from 22 candidate genes of Malaria. We have considered the ubiquitous Dunn Index as the cluster validity index (CVI) to validate the generated clusters. Our experiments reveal that VI-MCL produces a lot of singletons set on both synthetic graph and PPI graph of those 22 genes. MCL performs best among the three algorithms we have chosen. The quality of clusters produces by R-MCL is better than that of VI-MCL.**

Keywords* — *Graph-based Clustering; Dunn Index; Markov Chain; Gene Data; ; Protein-Protein-Interaction Network

I. Introduction

Clustering is a task where similar objects are grouped into the same groups and others into different groups. In statistical data analysis, clustering is used as a common technique in different fields. Some uses are in machine learning [1], computer graphics, bioinformatics, information retrieval [2], image analysis, and pattern recognition. There are so many clustering algorithms where data points are coming from Gaussian distribution like k-means [3] [1]. Togate the high-quality result we need to know some prior knowledge about clusterings like parameter or threshold value, the initial number of clusters, etc. To solve this problem, we can think of an algorithm that belongs to graph theory i.e., the Markov clustering algorithms where problems are categorized as Undirected Graph. In the Markov clustering, data are transformed into a graph representation [4]. Data points are the vertices of the graph to be clustered. The edges between the data points or the node are weighted by their similarity. Usually, the model does not require any prior knowledge however users provide some parameters with value. The Markov clustering methods are broadly used in biomedical and biological studies to determine the relationship among the objects [5]. We have used an interesting and efficient graph-based clustering algorithm in our experiment namely MCL, R-MCL, and VI-MCL. We have used both real data (Malaria genes) as well as synthetic datasets to evaluate and compare graph-based algorithms and analyses. The result of the analysis of all the clusters is shown in section V.

The rest of this paper is indexed as follows. Section II presents an overview of the literature survey on graph-based clustering. Section III presents the materials and methods. Section IV shows the experimental results. Section V presents discussions and conclusions.

II. Related Works

A graph-based clustering algorithm using non-Gaussian field data has been presented in [6]. The method has been applied to applications of computer vision and tested on two different data (Microcalcification Cluster Detection and News-Videos Segmentation). The author has proposed a Fuzzy c-means MST Clustering algorithm (FCM) which is based on cluster analysis.

The researcher has proposed graph clustering [7] and assessing the performance of different clustering algorithms namely: MCL, Iterative Conductance Cutting (ICC) and Geometric MST Clustering (GMC).

Applications of Ant Lion Optimization (ALO) and Cuckoo Search (CS) have been discussed on [8] protein-protein interaction for graph-based clustering. The researcher has used the R-MCL method on SARS-CoV-2 and the human dataset. The results indicate that CS-RMCL interactions are more stable than ALO-RMCL.

Researcher have proposed a study [9] on the PPI network on candidate genes of the Schizophrenia dataset. They have been implemented and simulated R-MCL graph-based clustering algorithm and the result is compared with the MCL algorithm on the same parameters.

The researcher has proposed the application of R-MCL method [10] and analyzed the similarity of the dengue virus. The clustering is done on the dengue virus of 30 protein sequences.

Researcher have used Stepping-stone type RMCL on Japanese associative concept dictionary and got a satisfactory level of performance than the Markov clustering algorithms generated network [11]. They have summarized the problems of MCL algorithms and proposed a

Stepping-stone type algorithm of R-MCL algorithms as an extension of the MCL Algorithm.

III. Method and Datasets

A. Graph-based Clustering

We have analyzed all the three graph-based clustering algorithms (MCL, R-MCL, and VI-MCL) and compared their experimental results.

1) Markov Clustering algorithm (MCL): Markov Clustering algorithm (MCL) [12] based on the idea of the random walk where dense clusters cannot be visited until most of the vertices are being visited. A transition probability of a matrix is iteratively called by the MCL. There is two main work done in MCL namely $expansion(k)$ and $inflation(r)$. Let M be the stochastic matrix of G (input matrix of MCL) then k is taken as the power of $k \in \mathcal{N}$ (natural numbers) of M where $\mathcal{N} > 1$. The purpose of inflation is to strengthen intra-cluster flows while weakening inter-cluster flows. In the time of inflation operation, entrywise powers are taken so that the output matrix is again stochastic. It must be noted that $r < 1$ defines the homogeneity of a row and heterogeneity probabilities of row emphasize by the value of $r > 1$. We have used 0.001 as the pruning threshold, $k = 6$ and $r = 3$. This clustering algorithm is used in bioinformatics for its effectiveness property and noise tolerance. MCL is simple and elegant but very slow. It produces too many clusters as output after pruning. The MCL pseudocode is given in Algorithm 1.

Algorithm 1: Markov Clustering(G, k, r)

Input: Let undirected graph $G = (V, E)$, k = expansion exponent, r = inflation exponent
Output: Cluster $\mathcal{C}$

begin
  Let $A(G)$ be the adjacency matrix and $D(G)$ be the diagonal matrix of degree of vertex.
  $A(G) \leftarrow A(G) + I_x$
  $M(G) \leftarrow A(G) \cdot D^{-1}(G)$
  $M \leftarrow M(G)$
  while M is not converge **do**
    calculate $M_e \leftarrow M^k$ `// expansion`
    calculate $M_i \leftarrow M_e^r$ `// inflation`
    $M_p \leftarrow M_i$ `// pruning on` M_i
    $M \leftarrow M_p$
  calculate $\mathcal{C}$ from M_p
  return $\mathcal{C}$

2) Regularized Markov Clustering (R-MCL) : R-MCL is the modification of MCL and developed by Parthasarathy and Satuluri in 2009 [13]. The main three steps of the algorithm are Regularization (regularized the input of the Markov matrix), Inflation (re-normalized), and Pruning. The value of the inflation parameter (usually 2) is given from the outside to get good results. Here, the inflation factor value is fixed to 3. Regularized Markov Clustering is completed in three-phase to cluster the protein sequence. In the first phase, matrix M is taken as an adjacency matrix and normalized to the adjacency matrix to form the input matrix of Markov clustering [13]. In R-MCL second phase, the regularized process has been done. The main aim of this process is to bring off protein interactions. Matrix multiplication is done in the regularized process. Let M_{col} is the initial input matrix i.e., Markov matrix input which is multiplied with matrix multiplication results that are the input of Markov matrix in each iteration. Initially, the input of the regularized matrix multiplication is Markov matrix input itself $M = M_{col}$ and the payoff matrix regularized (M_R) is performed as $M_R = M \times M_{col}$, and the result is stored as an $M = M_R$ (Markov matrix input) for the next iteration. In the third phase of R-MCL, inflation factor(r) is used to inflate, that is the process of power function execution of each element of M_R. Inflate preserved the initial topology of the graph as well as strong interaction got the strength and weak interaction become weaker. Now, again direct normalization is done on the results of regularized inflate matrix. By default value of the parameter, r is 2 and the matrix element becomes not uniform. Matrix holds a uniform element when the parameter value of r between 0 to 1. If the value of $r < 1$ then the small value becomes larger and the large element changed to smaller. Iteration should be continued until convergence of inflating matrix. In each iteration Regularization and Inflate are done and generate an idempotent matrix. The minimum threshold value of k is called the idempotent condition and the default value is $k = 10^-3$ [14]. The pseudocode of R-MCL is described in Algorithm 2.

Algorithm 2: Regularized MCL(G, r, M_{col})

Input: Let undirected graph $G = (V, E)$, r = inflation exponent, M_{col} = column stochastic matrix
Output: $\mathcal{C}$

begin
  $M \leftarrow M_{col}$
  Let $A(G)$ be the adjacency matrix and $D(G)$ be the diagonal matrix of degree of vertex.
  $A(G) \leftarrow A(G) + I_x$
  $M_{col} \leftarrow A(G) \cdot D^{-1}(G)$
  while M is not converge **do**
    regularize: $M_R = M \cdot M_{col}(G))$
    calculate $M_i = M_R^r$ `// inflation`
    $M_p \leftarrow M_i$ `// pruning on` M_i
    $M \leftarrow M_p$
  calculate $\mathcal{C}$ from M_p
  return $\mathcal{C}$

3) Variable Inflation MCL(VI-MCL): Constant inflation exponent is the disadvantage of R-MCL. The purpose behind the introduction of the variable rate of inflation factor is two-fold. First, the inflation rate must be kept maximal in the first stage. As a result that the information of the cluster begins quickly and thus the first stage does not last too long. And second, to obtain high-quality clusters we need a smaller inflation rate. The variable inflation exponent (r_{var}) for i-th iteration may be described as:

$$r_{var}(i) = 1 + r_0 \times exp(-\frac{i}{\tau})$$

The pseudocode of the VI-MCL algorithm is set out in Algorithm 3. We have used $r_0 = 5$, $\tau = 10$ experimentally. We have applied MCL and R-MCL to PPI networks of Malaria candidate genes data.

Algorithm 3: VARIABLE INFLATION MCL(G, k, r_{var})

Input: Let undirected graph $G = (V, E)$, k = expansion exponent, r_{var} = variable inflation exponent
Output: Cluster $\mathcal{C}$

begin
  Let $A(G)$ be the adjacency matrix and $D(G)$ be the diagonal matrix of degree of vertex
  $A(G) \leftarrow A(G) + I_x$
  $M(G) \leftarrow A(G) \cdot D^{-1}(G)$
  $M \leftarrow M(G)$
  while M is not converge **do**
    calculate $M_e \leftarrow M^k$ `// expansion`
    calculate $M_i \leftarrow M_e^{r_{var}}$ `// inflation`
    $M_p \leftarrow M_i$ `// pruning on` M_i
    $M \leftarrow M_p$
  calculate $\mathcal{C}$ from M_p
  return $\mathcal{C}$

B. Dataset

We have used Malaria gene datasets in our experiment and the dataset downloaded from the website Universal Protein Resource Knowledgebase (UniProtKB) which is freely available. These datasets have been collected from a database in uncompressed excel format and have 22 genes of Human. We perform clustering analysis on PPI networks using MCL, R-MCL, and VI-MCL graph-based clustering algorithms. We have used STRING to construct a PPI network, a well-known functional protein association network. The gene names with a description of data sets are shown in Table I.

C. Execution environments

We have implemented our experimental execution on a Lenovo ThinkPad E14 Ultrabook running the Windows 10 Professional 64-bit operating system and 10th Generation Intel Core i7-10510U Processor. The clock speed of the processor is 1.8 GHz with 16G bytes DDR4 memory size. The code has been executed in Python programming language (Version 3.6) in the Jupyter Notebook of Conda environment.

Table I. TWENTY TWO MALARIA GENES

	Description of Gene Data		
	Entry	***Entry Name***	***Gene Name***
1	P68871	HBB_HUMAN	HBB
2	P16671	CD36_HUMAN	CD36 GP3B GP4
3	P17927	CR1_HUMAN	CR1 C3BR
4	P31994	FCG2B_HUMAN	FCGR2B CD32 FCG2 IGFR2
5	P58753	TIRAP_HUMAN	TIRAP MAL
6	P05362	ICAM1_HUMAN	ICAM1
7	P01375	TNFA_HUMAN	TNF TNFA TNFSF2
8	P35228	NOS2_HUMAN	NOS2 NOS2A
9	Q16570	ACKR1_HUMAN	ACKR1 DARC FY GPD
10	O14931	NCTR3_HUMAN	NCR3 1C7 LY117
11	P04921	GLPC_HUMAN	GYPC GLPC GPC
12	P11413	G6PD_HUMAN	G6PD
13	Q08495	DEMA_HUMAN	DMTN DMT EPB49
14	P02724	GLPA_HUMAN	GYPA GPA
15	Q9NSE2	CISH_HUMAN	CISH G18
16	P02730	B3AT_HUMAN	SLC4A1 AE1 DI EPB3
17	P11277	SPTB1_HUMAN	SPTB SPTB1
18	P16284	PECA1_HUMAN	PECAM1
19	Q8TCT6	SPPL3_HUMAN	SPPL3 IMP2 PSL4
20	Q8TCT7	SPP2B_HUMAN	SPPL2B IMP4 KIAA1532
21	P16157	ANK1_HUMAN	ANK1 ANK
22	P35613	BASI_HUMAN	BSG UNQ6505/PRO21383

IV. ANALYSIS OF RESULTS

The Protein-Protein Interaction (PPI) network has been established to predict and analyze the function of protein in terms of physical interaction. The PPI network is a model that represents graphical connectivity between proteins. Though all the proteins are connected in the network there may also be some isolated components. We can present it as a graph where genes are represented as nodes and interactions are represented as an edge. We have used STRING (Search Tool for the Retrieval of Interacting Genes / Proteins) to construct a PPI network. Cytoscape platform has been used to visualize the networks.

We have analyzed our results and organized them (only two PPI networks and two randomly generated graphs) by the figure from Fig. 1 to Fig. 20. We have constructed PPI networks with vertex range 150 to 250 with non-uniform increment and five randomly generated graphs with vertex range 150 to 250 with a uniform increment of 22 genes of the dataset. Fig. 1 and Fig. 2 shows the PPI graph $G_1(V_1, E_1)$, $G_2(V_2, E_2)$ and Fig. 3 and Fig. 4 represents the random graph $G_3(V_3, E_3)$, $G_4(V_4, E_4)$ with $|V_1| = 154$, $|V_2| = 244$, $|V_3| = 150$ and $|V_4| = 250$. Output clusters of graphs are shown in Fig. 5 to Fig. 10. The iteration count versus execution time of four graphs (G_1, G_2, G_3, and G_4) are shown in Fig. 11 to Fig. 14. We have used the sparse matrix to do the experiment. The density of the matrix has been calculated in every iteration of the experiment. Fig. 15 to Fig. 18 indicates the sparseness of the graph (G_1, G_2, G_3, and G_4). We

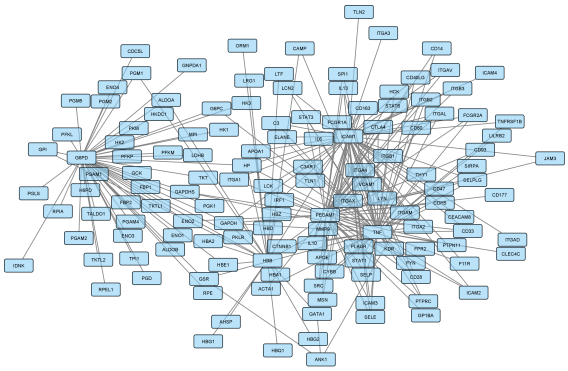

Figure 1. PPI Network G_1.

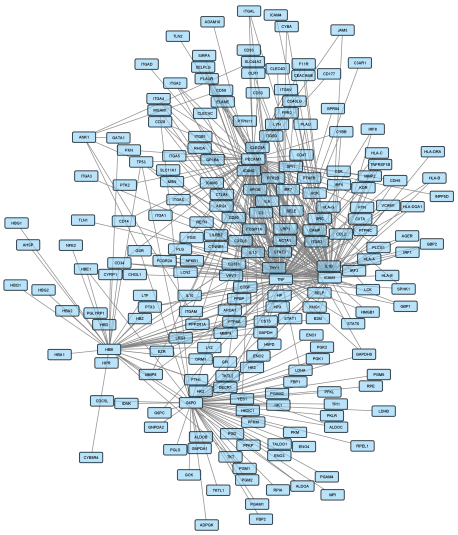

Figure 2. PPI Network G_2.

have validated our clustering using the Dunn index (DI) and the quality of the clustering is very magnificent. DI obtained from the PPI network and the random graphs visualized in Fig. 19 and 20. The analysis shows that the performance of MCL is good enough and R-MCL performs better than MCL. In a randomly generated geometric network, random points are generated and placed with a 2 dimension unit cube. We have taken distance threshold as the radius of 0.3 and distance metric as 2.

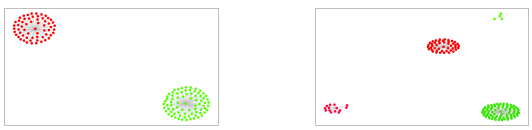

Figure 5. Cluster of G_1 (MCL and R-MCL)

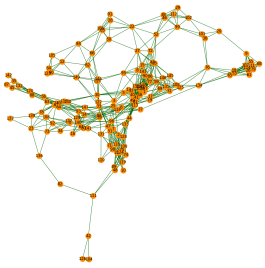

Figure 3. Random Network G_3.

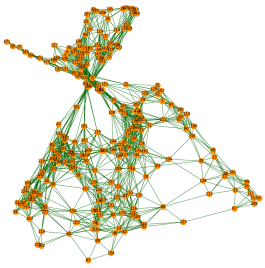

Figure 4. Random Network G_4.

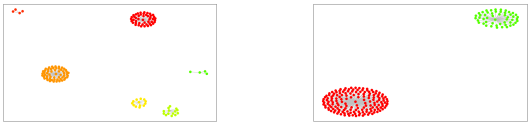

Figure 6. VI-MCL Cluster of G_1 and MCL Cluster of G_2

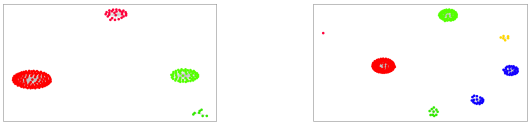

Figure 7. Cluster of G_2 (R-MCL and VI-MCL)

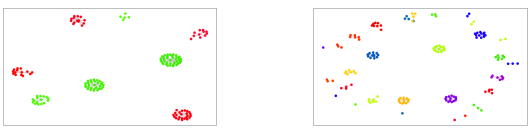

Figure 8. Cluster of G_3 (MCL and R-MCL)

[H]

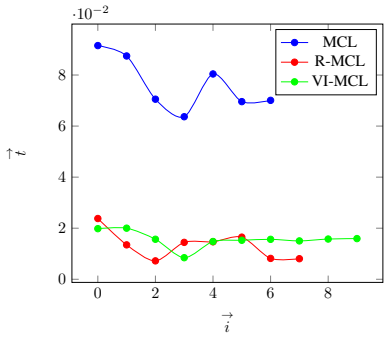

Figure 11. Execution time per iteration of G1

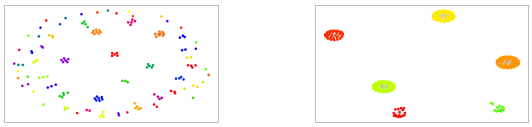

Figure 9. VI-MCL Cluster of G_3 and MCL Cluster of G_4

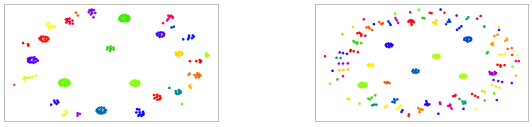

Figure 10. Cluster of G_4 (R-MCL and VI-MCL)

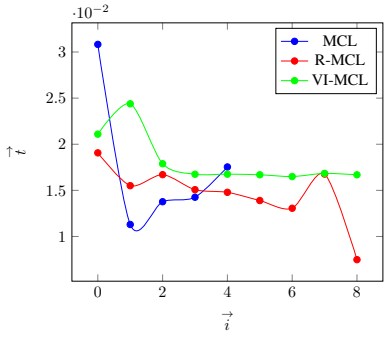

Figure 12. Execution time per iteration of G2

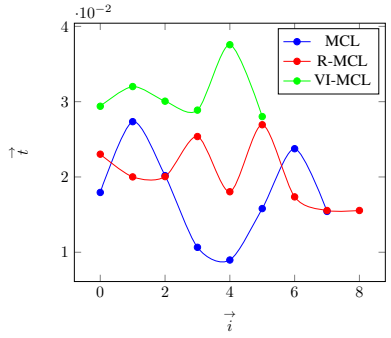

Figure 13. Execution time per iteration of G_3

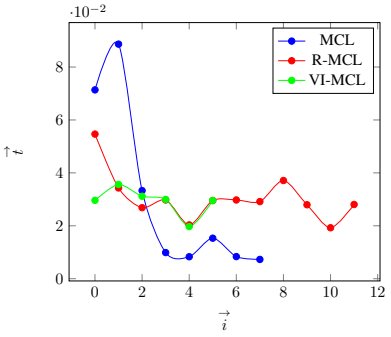

Figure 14. Execution time per iteration of G_4

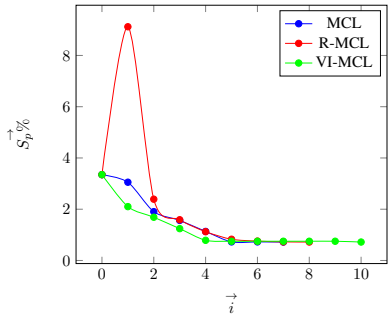

Figure 15. Matrix Sparseness of G_1

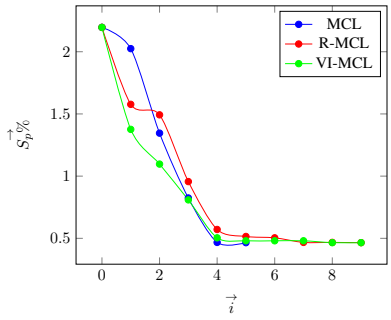

Figure 16. Matrix Sparseness of G_2

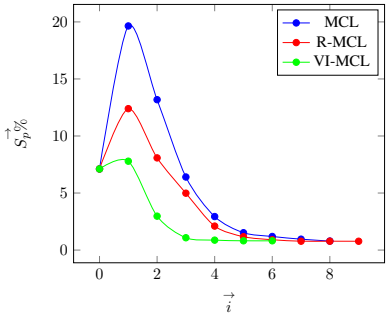

Figure 17. Matrix Sparseness of G_3

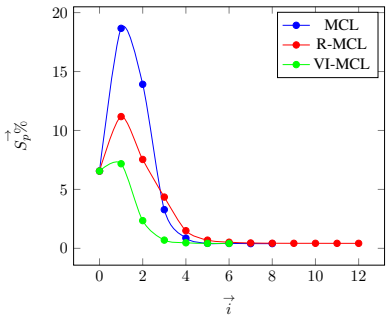

Figure 18. Matrix Sparseness of G_4

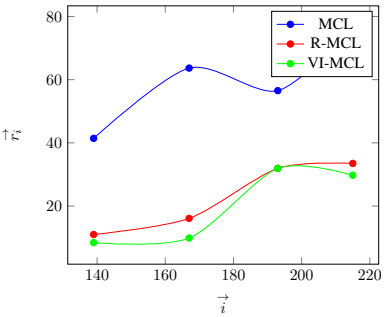

Figure 19. DI of PPI Network

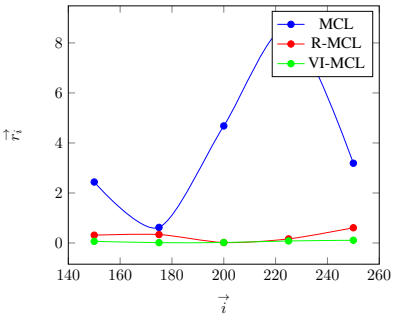

Figure 20. DI of Random Graph

V. Discussion and Conclusions

In this paper, we have analyzed graph-based clustering algorithms using Malaria genes with the help of MCL, R-MCL, and VI-MCL. We have validated our clustering using DI and the quality of the clustering is very magnificent. To evaluate our clustering algorithm, the DI metric is interpreted as an intra-cluster and inter-cluster distance. DI in the results section shows that the performance of the MCL is up to the mark. Performance of R-MCL is superior to VI-MCL. The study proposes that PPI on the Malaria candidate gene is extremely crucial for human disease. A cluster of one protein does not reveal much information. So the presence of more singleton protein in a cluster may be considered a bad cluster. As a consequence of this current study, protein family analysis of the draft human genome may be introduces based on the clustering results or measuring the protein similarities.

References

[1] S. Kamalanathan, S. Lakshmanan, and K. Arputharaj, "Enhanced K-means clustering algorithm for evolving user groups," *Indian Journal of Science and Technology*, vol. 8, no. 24, p. 1, 2015.

[2] M. Das, S. Kamalanathan, and P. Alphonse, "A Comparative Study on TF-IDF Feature Weighting Method and its Analysis using Unstructured Dataset," in *5th International Conference on Computational Linguistics and Intelligent Systems.*, 2020.

[3] A. K. Jain and R. C. Dubes, *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[4] S. G. Roy and A. Chakrabarti, "A novel graph clustering algorithm based on discrete-time quantum random walk," in *Quantum Inspired Computational Intelligence*. Elsevier, 2017, no. 361–389.

[5] Y. Zhang, Z. Ouyang, and H. Zhao, "A statistical framework for data integration through graphical models with application to cancer genomics," *The annals of applied statistics*, 2017.

[6] P. Foggia, G. Percannella, C. Sansone, and M. Vento, "A graph-based clustering method and its applications," in *International Symposium on Brain, Vision, and Artificial Intelligence*, 2007.

[7] P. Foggia and G. Percannella, "Assessing the performance of a graph-based clustering algorithm," in *International Workshop on Graph-Based Representations in Pattern Recognition*, 2007.

[8] A. Rizki, Bustamam, and D. Sarwinda, "Applications of cuckoo search and ant lion optimization for analyzing protein-protein interaction through regularized Markov clustering on coronavirus," in *Journal of Physics: Conference Series*, 2021.

[9] R. Ginanjar, A. Bustamam, and H. Tasman, "Implementation of regularized Markov clustering algorithm on protein interaction networks of schizophrenia's risk factor candidate genes," in *2016 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2016, pp. 297–302.

[10] D. Lestari, D. Raharjo, A. Bustamama, B. Abdillah, and W. Widhianto, "Application of clustering methods: Regularized Markov clustering (R-MCL) for analyzing dengue virus similarity," in *AIP Conference Proceedings*, vol. 1862, no. 1, 2017, p. 030130.

[11] J. Jung, M. Miyake, and H. Akama, "Recurrent Markov Cluster (RMCL) Algorithm for the Refinement of the Semantic Network." in *LREC*, 2006, pp. 1428–1431.

[12] V. Dongen and S. Marinus, "Graph clustering by flow simulation," Ph.D. dissertation, 2000.

[13] V. Satuluri, S. Parthasarathy, and D. Ucar, "Markov clustering of protein interaction networks with improved balance and scalability," in *ACM*, 2010, pp. 247–256.

[14] A. Bustamam, K. Burrage, and N. A. Hamilton, "Fast Parallel Markov Clustering in Bioinformatics Using Massively Parallel Computing on GPU with CUDA and ELLPACK-R Sparse Format," 2012.