A Parallel Algorithm to Generate Connected Network Motifs

Efendi Zaenudin, Ezra Bernadus Wijaya†, Eskezeia Yihunie Dessie†, Mekala Venugopala Reddy†, Jeffrey J.P. Tsai, Chien-Hung Huang*, Ka-Lok Ng*

Abstract—Network of interactions among bio-molecules is fundamental to biological processes. Many works have shown that molecular networks can be analyzed by decomposing the networks into smaller modules named network motifs. We hypothesize that identifying the set of possible 5-node motifs embeds in a network is a necessary step to elucidate the complex topology of a network. To achieve this goal, it requires to determine the complete set of motifs that are compose of five connected nodes.

We developed an algorithm to remove motifs compose of disconnected components and implemented a parallelized algorithm to reduce the computation time. Our experiment demonstrated that the proposed parallel algorithm is approximately 1.3 times faster than serial programming for identifying 5-node motifs with all the nodes connected.

***Index Terms* -** Molecular Networks; Graph theory; Digraphs; Network Motifs; Isomorphic Graphs; Permutation Matrices

I. Introduction

Molecular networks are composed of genetic elements that exhibit activation or suppression mechanisms. In the post-genome era, it is more productive to investigate how bio-molecules regulate or cooperate on a system level. Decomposing a complex network into a number of small module is a useful concept in elucidating the underlying network topological structures. Such modules are called network motifs. Examples of such motifs are auto-regulation (either catalytic or repression), coherent feed-forward loop, single-input module and bi-fan [1-3]. Network motifs play an important role in various types of networks, including molecular networks, ecological networks, electrical networks etc. [4-11]. Motifs comprise of *N* nodes are referred to as '*N*-node motifs' in this paper. It is known that the time complexity of identifying *N*-node motifs in a large network is a NP-complete problem [12]. We note that there was a work claimed that network motifs do not necessary determine biological functions, there is no characteristic behavior for network motifs [13], while other works [14-16] reported opposite results. Multilayer network description is also an important area in multi-omics study, the concept of: (i) tensorial framework has been introduced to study this problem [17], and (ii) graph isomorphism has been generalized to determine whether two multilayer networks equivalent structurally or not [18].

The work of Chien-Hung Huang is supported by the grant of Ministry of Science and Technology of Taiwan (MOST) MOST 107-2221-E-150-038. The work of Efendi Zaenudin, Ezra Bernadus Wijaya, Eskezeia Yihunie Dessie, Mekala Venugopala Reddy and Dr. Ka-Lok Ng are supported by MOST under the grant of MOST 106-2221-E-468-017. Dr. Ka-Lok Ng work is also supported by the grants of MOST 108- 2221-E-468-020, 107-2632-E-468-002, 107-2221-E-150-038 and grants from Asia University, 107-asia-02 and 107-asia-09. Dr. Jeffrey J. P. Tsai work is supported by the grant of MOST 107-2632-E-468-002.

Efendi Zaenudin, Ezra Bernadus Wijaya, Eskezeia Yihunie Dessie and Mekala Venugopala Reddy are with the Department of Bioinformatics and Medical Engineering, Asia University, they are supported by supported by MOST 107-2632-E-468-002. Ezra Bernadus Wijaya, Eskezeia Yihunie Dessie and Mekala Venugopala Reddy equally contributed to this work. Efendi Zaenudin is with Research Center for Informatics, Indonesian Institute of Sciences, Bandung, Indonesia.

*Corresponding author, Chien-Hung Huang is with the Department of Computer Science and Information Engineering, National Formosa University (e-mail: chhuang@nfu.edu.tw). *Corresponding author, Ka-LokNgis with the Department of Bioinformatics and Medical Engineering, Asia University & Department of Medical Research, China Medical University Hospital, China Medical University, Taiwan (e-mail: ppiddi@gmail.com).

We extended our IMECS2019 conference paper [19] by developing an algorithm to remove motifs consist of disconnected components, and generate a complete set of motifs that are compose of five connected nodes. After excluding motifs consist of isolated node(s) or pattern(s) and take into account of isomorphic motifs, a total of 13, 199, and 9364 possible motif patterns can be defined for the 3-node, 4-node and 5-node motifs, respectively [20-21]. Also, we implemented a parallelized algorithm to reduce the computation time for generating the 5-node motifs. Our experiment demonstrated that the parallelized algorithm is approximately 1.3 times faster than serial programming.

II. Methods

Let *G* be a graph or motif and *G* = (***V***, ***E***), where *V* and *E* denote vertices and edges. An adjacency matrix, ***A***, can be constructed to represent the connectivity of each node embeds in the motif. The matrix ***A*** is composed of matrix elements with a value of either '0' or '1', which denote non-connected or connected nodes, respectively. In this work, we do not consider self-interacting node. Therefore, the diagonal elements of ***A*** are zeros. For *N*-node motifs, the possible number of edges range from $N-1$ to $2*C(N,2)$ edges.

Given the adjacency matrix, one can denote a motif by an integer. For example, the 3-node motif named 'cascade' can be represented by the binary string 000001100, which is equivalent to '12' in decimal representation. The complete set of the *N*-node motifs can be represented by 2^X adjacency matrices, where $X = 2*C(N, n)$, and $C(N, n)$ is the combinatorial factor for choosing *n* nodes from *N* nodes. The factor of two arises because we consider digraph.

A motif may consist of isolate node(s) that doesn't connect to the other nodes. In the 4- and 5-node cases, motif with disconnected pattern(s) and isolated node(s) are illustrated in Table 1. Disconnected patterns are shown in the second and third rows of Table 1; whereas the rest of the rows are motifs compose of disconnected nodes. Both types of motifs were removed in this study.

We noted that one cannot identify any path to connect isolated node(s) or disconnected pattern(s); therefore, motifs compose of disconnected components can be identified by working with the matrix ***A***. For an undirected graph, given nodes *i* and *j*, the number of paths that connect nodes *i* and *j* can be determined by forming matrix product of the adjacency matrix. Since we are working with directed graph, so matrix ***A*** is asymmetric, and a symmetric matrix ***B*** can be defined by Eq. (1),

$$[B = A + A^T],\ [B^2, B^3, \dots] \tag{1}$$

where $\boldsymbol{A}^T$ denotes the transpose of ***A***, and $\boldsymbol{B}^2$ and $\boldsymbol{B}^3$ denote the 2^{nd} and 3^{rd} power of ***B*** respectively. The matrix element b_{ij} (i not equals to j) for $\boldsymbol{B}^n$ represents the total number of the shortest paths of length n between vertices i and j in a graph G. If any b_{ij} equals to zero, it means disconnected graph, then it will be removed; hence, we are able to identify the complete set of motifs compose of five connected nodes.
Table 2 is an illustration of five isomorphic structures that compose of five nodes and four edges. The five isomorphic motifs are associated with decimal numbers 30, 928, and 27648, 753664 and 15728640. Further analysis shows that adjacency matrices associated with isomorphic graphs are related by permutation matrix multiplication.

TABLE 1
ISOLATED MOTIFS AND DISCONNECTED MOTIFS

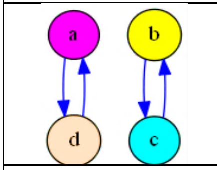
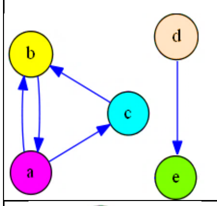
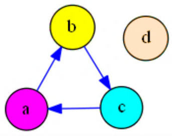
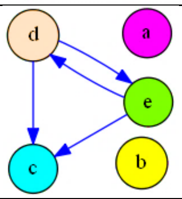
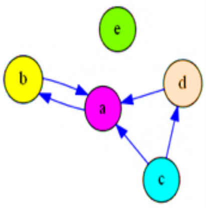
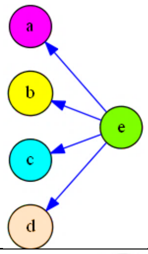
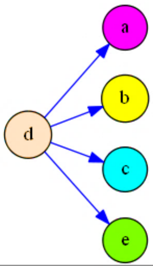
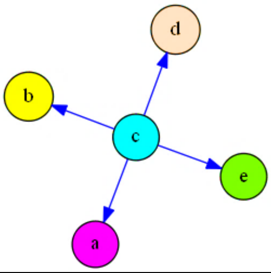
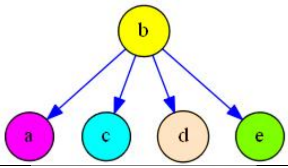
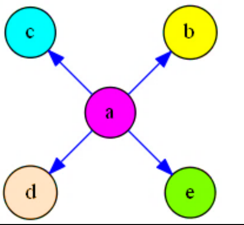
Network Motif	Node	Edges	Adjacency Matrix	Decimal Number
	4	4	$\begin{bmatrix}0&0&0&1\\0&0&1&0\\0&1&0&0\\1&0&0&0\end{bmatrix}$	4680
	5	5	$\begin{bmatrix}0&1&1&0&0\\1&0&0&0&0\\0&1&0&0&0\\0&0&0&0&1\\0&0&0&0&0\end{bmatrix}$	13115424
	4	3	$\begin{bmatrix}0&1&0&0\\0&0&1&0\\1&0&0&0\\0&0&0&0\end{bmatrix}$	17024
	5	4	$\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&1&0&1\\0&0&1&1&0\end{bmatrix}$	166
	5	5	$\begin{bmatrix}0&1&0&0&0\\1&0&0&0&0\\1&0&0&1&0\\1&0&0&0&0\\0&0&0&0&0\end{bmatrix}$	8931840

TABLE 2
ISOMORPHIC NETWORK MOTIFS COMPOSE OF FIVE NODES AND FOUR EDGES

Network Motif	Adjacency Matrix	Decimal Number
	$\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\1&1&1&1&0\end{bmatrix}$	30
	$\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\1&1&1&0&1\\0&0&0&0&0\end{bmatrix}$	928
	$\begin{bmatrix}0&0&0&0&0\\0&0&0&0&0\\1&1&0&1&1\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$	27648
	$\begin{bmatrix}0&0&0&0&0\\1&0&1&1&1\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$	753664
	$\begin{bmatrix}0&1&1&1&1\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\\0&0&0&0&0\end{bmatrix}$	15728640

Motifs with isomorphic structures are related by matrix multiplication. For 4-node motifs, the permutations are given by 24 permutation matrices including the identity matrix. For 5-node motifs, there are 120 permutation matrices. In general, there are ***N***! permutation matrices, it is because each row and column of the permutation matrix consists of only one "1". For 4-node motifs, let P_k denotes the permutation matrix, where k equals 0 to 23, the 24 matrices are given by:

$$P_0 = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}, P_1 = \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}, P_2 = \begin{bmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{bmatrix},$$

$$P_3 = \begin{bmatrix}1&0&0&0\\0&0&1&0\\0&0&0&1\\0&1&0&0\end{bmatrix}, P_4 = \begin{bmatrix}1&0&0&0\\0&0&0&1\\0&1&0&0\\0&0&1&0\end{bmatrix}, P_5 = \begin{bmatrix}1&0&0&0\\0&0&0&1\\0&0&1&0\\0&1&0&0\end{bmatrix},$$

$$P_6 = \begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&1&0\\0&0&0&1\end{bmatrix}, P_7 = \begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}, P_8 = \begin{bmatrix}0&1&0&0\\0&0&1&0\\1&0&0&0\\0&0&0&1\end{bmatrix},$$

$$P_9 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P_{10} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_{11} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$
$$P_{12} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, P_{13} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{14} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$
$$P_{16} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P_{16} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{17} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix},$$
$$P_{18} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}, P_{19} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{20} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$
$$P_{21} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}, P_{22} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}, P_{23} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \quad (2)$$

The first matrix denoted by P_0 is the identity matrix. Some of the permutation matrices are symmetric, such as, P_0, P_1, P_2, and P_5; some are asymmetric (trace $(P_i) = 0$), i.e. P_9, P_{10} and P_{13}. The inverse of a permutation matrix is again a permutation matrix, i.e. $P^{-1} = P^T$.

Left multiplication of adjacent matrix, $P_L A$ resulted in row interchange, whereas right multiplication of A, i.e. AP_R, resulted in column interchange. It is known that matrices multiplication satisfies the association law, i.e.$P_L(AP_R) = (P_L A)P_R$. If $P_L(AP_R) = X$, then motif A and X are motifs with isomorphic structures. Since the indices L and R run from 0 to 5, there are a total of 36 row and column interchanging operations.

One can consider matrix multiplication of two permutation matrices, i.e. $P_y P_x$; however, product of permutation matrices equals to a permutation matrix; such as, $P_1 P_2 = P_4$ and $P_5 P_2 = P_3$. Hence, there is no need to consider higher order matrix multiplications.

Figure 1 is the workflow of our algorithm to identify 5-node motifs compose of five connected nodes. Table 3 is the description of the workflow shown in Figure 1. Below we provide the pseudo-codes of the workflow steps.

TABLE 3
THE SEVEN STEPS OF THE WORKFLOW

Step1–Generate All Possible Number of Edges for *N*-Node Motifs, i.e. from *N-1* to 2**C(N,2)* Edges.
Step2–Generate All Possible Permutation Matrices.
Step3–Generate All Possible Adjacency Matrices and denote each Matrix by a Decimal Number.
Step4–Classify The Decimal Number Representation According to The Number of Edges.
Step5–Isomorphic Patterns are Identified by Permutation Matrices Multiplication.
Step6–Isomorphic Patterns are Removed Except The One Associates with The Minimal Decimal.
Step7–Remove All Disconnected Patterns Using The 4th Power of Matrix A

Step 1: Generate all possible patterns with N-1 to 2*C(N,2) edges

Input: *integer N*
Output: *[minimum-edge, ..., maximum-edge]*

```
maximum-edge=N^2 − N
minimum-edge=N-1
while minimum-edge<=maximum-edge do
    edge-index=[minimum-edge++]
```

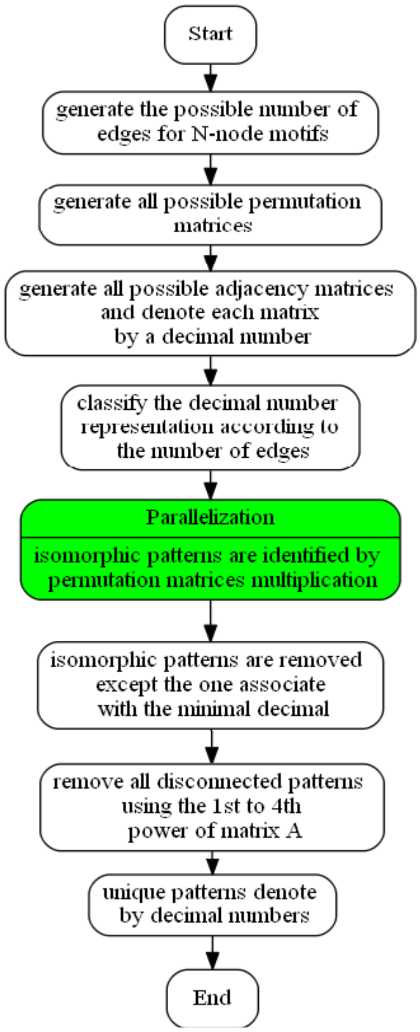


FIGURE 1. WORKFLOW OF THE PRESENT STUDY

Step 2: Generate all possible permutation matrices

Input: *identity_matrix_permutation*
Output: *permutation_matrix*

```
temp_matrix=identity_matrix_permutation
Call multiset_permutation
Function save_numberof_permutation=
multiset_permutation.sum()
for save_numberof_permutation in
  multiset_permutation(list_permutation) do
    save_list_permutation(save_numberof_permutation
```

Step 3: Generate all possible adjacency matrices and denote each matrix by a decimal number

Input: *integer N*
Output: *A set of decimal numbers represent motifs with at least one edge*

```
matrix_adjccy=numpy.ones// change diagonal to zero
for i in reversed(matrix_adjccy(element)) do
    max_decimal_number=2^(counterxint(i))
    counter+=1
for i in range(2, max_decimal_number) do
    save_number_of_decimal=i
    i+=1
```

```
while i<length(save_numberof_decimal) do
    graph_inmatrix =numpy.binary_repr(
     save_number_of_decimal[i], width=NxN)
     graph_inmatrix_adj=numpy.array(graph_inmatrix)
    while j<length(diagonal_matrix_NxN) do
        if diagonal_matrix_NxN(graph_inmatrix_adj)=0
         then
            number_of_decimal[i]=
            save_number_of_decimal[i]
        j+=1
    i+=1
```

Step 4: Classify the decimal number representation according to the number of edges

Input: *A set of decimal numbers represent motifs with at least one edge*

Output: *Groups of decimals associate motifs with motifs compose of different number of edges, i.e. 4 to 20 edges for 5-node motifs*

```
while i<length(number_of_decimal) do
    graph_inmatrix=numpy.binary_repr
    (number_of_decimal[i],width=NxN)
    sum_of_element[i]=numpy.sum
    (graph_inmatrix[element])
    i+=1
while j<length(edge-index) do
    while k<length(number_of_decimal) do
        if sum_of_element[k]=edge-index[j] then
            class_of_decnum.extend([number_of_decimal])
        k+=1
    j+=1
```

Step 5: Isomorphic patterns are identified by permutation matrices multiplication

Input: *Groups of decimals associate motifs with motifs compose of different number of edges*

Output: *Groups of decimals correspond to isomorphic patterns*

```
while i<length(class_of_decnum) do
    decnum_classes =class_of_decnum[i]
    while j<length(decnum_classes) do
        adj_matrix_trf =numpy.binary_repr
        (decnum_classes[j],width=NxN)
        adj_matrix=numpy.array(adj_matrix_trf)
        while k<length(save_list_permutation* do
            permutation=save_list_permutation[k]
            permutation_transpose=permutation.transpose()
            matrix_isomorphm=((permutationxadj_matrix)
            xpermutation_transpose)
            Parallelization
            [decnum_isomorphm=convert
            (matrix_isomorphm)]
            k+=1
        collect_of_decnum_isomorphm.extend
        (decnum_isomorphm)
        j+=1
    class_of_decnum[i]=collect_of
    _decnum_isomorphm
    sorted(class_of_decnum[i])
    i+=1
```

**save_list_permutation is call permutation in step 3*

Step 6: Isomorphic patterns are removed except the one associate with the minimal decimal

Input: *Groups of decimals correspond to isomorphic patterns*

Output: *The minimal decimals, including motifs compose of disconnected components*

```
while i<length(class_of_decnum) do
    decnum_classes =classof_decnum[i]
    while j<length(decnum_classes) do
        decnum_minimal=decnum_classes[0]
         collect_of_decnum_minimal=decnum_minimal
         j+=1
    class_of_decnum=collect_of_decnum_minimal
    sorted(class_of_decnum)
    i+=1
```

Step 7: Remove all disconnected patterns using power matrices

Input: *The minimal decimals, including motifs compose of disconnected components*

Output: *The minimal decimals that represent motifs compose of connected nodes, i.e. connected motif patterns*

```
while i<length(class_of_decnum) do
    decnum_classes =class_of_decnum[i]
    while j<length(decnum_classes) do
        adj_matrix_trf =numpy.binary_repr
        (decnum_classes[j],width=NxN)
        adj_matrix=numpy.array(adj_matrix_trf)
        B=adj_matrix+(adj_matrix.transpose)
        variable_to_sum = 0
        while h<N do
            B2=matriXirtam(B2,B)
            B_listed(B2[0])
            h+=1
        while k<N do
            while l<N do
                variable_to_sum=variable_to_sum+
                B_listed[l][k]
                if l=minimum-edge AND
                 variable_to_sum=0 then
                    del_of_decnum=decnum
                l+=1
            k+=1
        j+=1
    class_of_decnum=class_of_decnum[i]-del_of_decnum
    i+=1
sorted(class_of_decnum)
```

Step 1 creates an index parameter to record the set of possible number of edges for *N*-node motifs, i.e. from *N-1* to 2**C(N,2)* edges. The "while" loop creates the edge-indexed list.Isomorphic motifs are related by permutation matrix multiplication.

Step 2 is to generate the set of all possible permutation matrices, i.e. *N!* for *N*-node motifs. This is done by calling the library 'multiset_permutation' (available in Python 3.6).

As step 3 shown, one defines a maximum decimal number for a given number of edges. We generate an adjacency matrix consists of the number of occurrence of '1' which equal to the number of edges. Then, we denote each matrix by a decimal number, and save the decimal numbers in a list.

In step 4, we grouped motifs with the same number of edges (without considering edge direction) together into the same list.

Step 5 is to group together all the isomorphic motifs that are related by permutation matrix multiplication. For instance, the group [6, 40, 192] means that 'motif_6', 'motif_40' and 'motif_192' are isomorphic. These three motifs are related by permutation matrix multiplication. Group, [12, 34, 66, 96, 132, 136], represents another group of isomorphic motifs, it has the same number of edges as group, [6, 40, 192]. Output from step 5 serves as the input for step 6.

After that, for each group of isomorphic motifs, step 5 selected the minimal decimal number to represent that group, which is shown as underline and bold-face font below.

Finally, we removed all the disconnected patterns using the power of matrix $\boldsymbol{B}$. A motif pattern was removed if the sum of the matrix element b_{ij} obtained from the 1^{st} to the 4^{th} power of $\boldsymbol{B}$ is equal zero.

III. Experiment

This aim of this section is to determine the speed up ratio after we parallelized the algorithm.

A. Computer environment

We implemented our algorithm in Python 3.6. Experiments are performed on two computers: (i) using Windows 10, Intel Core™ i5-6400 CPU @2.70 GHz CPU with 4-cores and 16 GB memory associated with CUDA - NVIDIA GeForce GT 720, and (ii) using Ubuntu 18.04, Intel Core™ i7-3770 CPU @3.40 GHz @3.90 GHz and 12 GB memory.

B. Comparison of speed up time

The results of the CPU execution time and speed up ratios for generating 5-node motifs are given in Table 4. As shown in Table 4, compare to machine 2, the algorithm utilized less CPU time than machine 1; i.e. about 2.4 times faster, that is supported by GPU-card. Also, the speed up ratio is around 1.3 times faster than serial programming.

IV. Results

In Table 5, we listed the total numbers of 3-node motifs, 4-node motifs and 5-node motifs. Our results agree with the numbers reported in Refs. [20-21]; hence, support the correctness of our algorithm.

A. Decimal representation of the 5-node motifs

In the Appendix section, Table A1 listed the three sets of decimals associate with the 3-node motifs, 4-node motifs and 5-node motifs. The total number of 5-node motifs is 9364, because of space limitation, only a partial list of the decimals is shown.

TABLE 4
SPEED UP SERIAL AND PARALLEL ON DIFFERENT COMPUTERS

Five Nodes	(hh:mm:ss)		Speed Up
	Serial	Parallel	
Machine 1	0:46:33.079	0:35:37.562	1.31
Machine 2	1:53:11.560	1:25:32.618	1.32
Speed Up	2.432	2.401	

TABLE 5
NUMBERS OF MOTIFS IDENTIFIED BY THE PRESENTED ALGORITHM

Node	3	4	5
Numbers of motifs found	13	199	9364

V. Conclusions

We have developed a systematic and rigorous method to generate the complete sets of 3-node, 4-node and 5-node network motifs without disconnected and isomorphic patterns. In addition, a parallelized version of our algorithm was designed to speed up the time for 5-node motifs identification, i.e. approximately 1.3 times faster. With the 5-node motif patterns available, we plan to identify the 5-node motifs embed in the molecular networks in the next stage; thus allow us to dissect the underlying topology structures of a network.

Appendix

Table A1. THE SET OF DECIMALS ASSOCIATES WITH THE 5-NODE MOTIFS.

Number of Edges	Motif ID [Decimal representation]
4	[30, 60, 154, 156, 184, 404, 1080, 1176, 1298, 1304, 1328, 1424, 1800, 3344, 8466, 8472, 8496, 8728, 33840, 33936, 34306, 34320, 35088, 35344, 41488, 49680, 541200] Total Number is 27
5	[62, 158, 186, 188, 406, 412, 1082, 1178, 1180, 1208, 1302, 1306, 1308, 1330, 1332, …, 1082402, 1082498, 1082500, 1082504, 1083522, 1083650, 1083652, 1083656, 1083664, 1090050, 1122820] Total Number is 108
6	[190, 414, 438, 444, 924, 1086, 1182, 1210, 1212, 1310, 1334, 1338, 1340, 1430, 1434, 1436, 1458, 1460, 1464, …, 1122480, 1122692, 1122822, 1122828, 1122836, 1122864, 1123848] Total Number is 326
7	[446, 926, 1214, 1342, 1438, 1462, 1466, 1468, 1822, 1838, 1850, 1852, 1934, 1946, 1948, …, 1139080, 1139210, 1139212, 1139224, 1150466, 1156610, 1157634, 1163556, 1163652, 1163908, 1164802] Total Number is 667
8	[958, 1470, 1854, 1950, 1966, 1978, 1980, 3262, 3390, 3486, 3510, 3514, 3516, 3870, ..., 1255948, 1255954, 1255984, 1262214, 1262218, 1262220, 1262228, 1263144, 1263152, 1598218, 1598232, 1623600] Total Number is 1127
9	[1982, 3518, 3902, 3998, 9150, 9662, 9918, 10046, 10142, 10158, 10166, 10170, ..., 1624852, 1624856, 1630982, 1630986, 1630994, 1631000, 1657478, 1657482, 1657484, 1657490, 1657492, 1657496, 1663622, 1663634] Total Number is 1477
10	[4030, 10174, 11710, 11966, 12094, 12190, 25534, 26430, 26526, 26556, 34750, ..., 3320472, 3320596, 3327270, 3327274, 3327366, 3327370, 3327378, 3327380, 3327384, 3327750, 3327754, 3328390, 3328774, 3328780, 3727750] Total Number is 1665
11	[12222, 26558, 36798, 42942, 43966, 44478, 44734, 44862, 44958, 44974, ..., 3392902, 3392906, 3392914, 3393158, 3393162, 3393170, 3399846, 3399852, 3399860, 3727758, 3727766, 3727788, 3727796, 3728786] Total Number is 1489
12	[28606, 44990, 53182, 59326, 60350, 61358, 61366, 61370, 102334, 108478, …, 3729290, 3729710, 3729722, 3729806, 3729814, 3729818, 3729836, 3729844, 3730220, 3730730, 3730834, 3731210, 3788178, 3788338, 3794322, 3826094] Total Number is 1154

13	[61374, 110526, 118718, 124862, 126398, 126654, 126782, 126878, 241086, 241342, …, 3794586, 3794604, 3794610, 3794612, 3794732, 3794738, 3794740, 3796370, 3796530, 3826110, 3827118, 3827130, 3827502, 3827514, 3859890, 3860146] Total Number is 379
14	[126910, 241598, 249790, 255934, 585662, 643006, 649150, 651070, 651166,…, 3860910, 3860918, 3860922, 3860924, 3861174, 3861178, 3861180, 3861302, 3861308, 3861404, 3861938, 3861944, 3862194, 3862200, 3862286, 3862322, 3927602] Total Number is 379
15	[257982, 651198, 1175486, 1273790, 1290174, 1298366, 1304510, 1305534, 1306542, 1306550, 1306554, 1634238, …, 3927606, 7581102, 7581110, 7581114, 7581370, 7581498, 7581500, 7581596, 7595930, 7974318, 7974326, 7974332, 7988654, 7989134] Total Number is 154
16	[782270, 1306558, 1699774, 1798078, 1830830, 1830838, 3272638, 3338174, 3387326, 3395518, 3401662, 3403198, …, 7595934, 7595964, 7974334, 7974814, 7988670, 7989150, 7989174, 7989178, 7989180, 7990966, 7990970, 7991098, 7991100, 8113846] Total Number is 61
17	[1830846, 3403710, 3796926, 3862462, 3927486, 3927742, 3927966, 7581630, 7595966, 7974846, 7989182, 7990974, 7991102, 7991198, 8113854, 8114078] Total Number is 16
18	[3927998, 7598014, 7991230, 8114110, 8120254] Total Number is 5
19	[8122302] Total Number is 1
20	[16510910] Total Number is 1
The total number of decimal numbers associate with the 5-node motifs is 9364.	

REFERENCES

[1] U. Alon, An introduction to systems biology – design principles of biological circuits, 1st edition. Chapman & Hall/CRC, 2007.

[2] Dardel Frederic and Kepes Francois. Bioinformatics – Genomics and post-genomics, 1st edition. Translated by Hardy Noah. Wiley, 2006.

[3] Avi Ma'ayan et al., "SNAVI: Desktop application for analysis and visualization of large-scale signaling networks". BMC Systems Biology, 3,10, 1-11, 2009.

[4] U. Alon, "Network motifs: theory and experimental approaches," *Nature Reviews Genetics*, vol. 8, no. 6, p. 450, 2007.

[5] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon, "Network motifs in the transcriptional regulation network of escherichia coli," *Nature genetics*, vol. 31, no. 1, p. 64, 2002.

[6] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: simple building blocks of complex networks," *Science*, vol. 298, no. 5594, pp. 824–827, 2002.

[7] J. Fodor, M. Brand, R. J. Stones, and A. M. Buckle, "Intrinsic limitations in mainstream methods of identifying network motifs in biology," *bioRxiv*, p. 272401, 2018.

[8] M. B. Z. Joveini and J. Sadri, "Application of fractal theory on motifs counting in biological networks," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 15, no. 2, pp. 613–623, 2018.

[9] K. Mukherjee, M. M. Hasan, C. Boucher, and T. Kahveci, "Counting motifs in dynamic networks," *BMC systems biology*, vol. 12, no. 1, p. 6, 2018.

[10] J. Wang, Y. Huang, F.-X. Wu, and Y. Pan, "Symmetry compression method for discovering network motifs," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 9, no. 6, pp. 1776–1789, 2012.

[11] K. A. Zweig, C. Brugger, V. Grigorovici, C. De Schryver, and N. Wehn, "Automated determination of network motifs," Jun. 7 2018, uS Patent App. 15/579,518.

[12] W. Kim, M. Diko, & K. Rawson, "Network motif detection: algorithms, parallel and cloud computing, and related tools". *Tsinghua science and technology*. 18(5), 469-489, 2013.

[13] S. Mangan, U. Alon, "Structure and function of the feed-forward loop network motif". Proceedings of the National Academy of Sciences of the United States of America 2003, 100:11980-11985.

[14] P.J. Ingram, M.P. Stumpf, J. Stark, "Network motifs: structure does not determine function". BMC genomics 2006, 7:108.

[15] M.M. Babu, N.M. Luscombe, L. Aravind, M. Gerstein, S.A. Teichmann, "Structure and evolution of transcriptional regulatory networks". Current opinion in structural biology 2004, 14:283-291.

[16] G. Swiers, R. Patient, M. Loose, "Genetic regulatory networks programming hematopoietic stem cells and erythroid lineage specification". Developmental biology 2006, 294:525-540.

[17] De Domenico Manlio, Solé-Ribalta Albert, Cozzo Emanuele, Kivelä Mikko, Moreno Yamir, Porter Mason A., Gómez Sergio, and Arenas Alex, "Mathematical Formulation of Multilayer Networks" Phys. Rev. X 3, 041022, 2013.

[18] M. Kivela & M. A. Porter. "Isomorphisms in Multilayer Network". *IEEE Transactions on Network Science and Engineering*, 5(3), pp. 198-211, 2018.

[19] Efendi Zaenudin, Ezra B. Wijaya, Eskezeia Y. Dessie, David Agustriawan, Jeffrey J.P. Tsai, Chien-Hung Huang, and Ka-Lok Ng, "An Algorithm to Construct Network Motifs for Bioinformatics Study," *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2019, 13-15 March, 2019, Hong Kong*, pp45-50.

[20] F. Harary and E. Palmer, "Graphical enumeration, acad," *Press, NY*, 1973.

[21] N. J. A. Sloane & S. Plouffe, *The Encyclopedia of Integer Sequences.* Academic Press, San Diego, 587, 1995.