IOP Conference Series: Materials Science and Engineering

PAPER • **OPEN ACCESS**

Searching Method of Structural Similar Subnets in Protein-protein Interaction Networks Based on Quantum Walks

To cite this article: Li-Ping Yang and Song-Feng Lu 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **490** 062021

View the article online for updates and enhancements.

This content was downloaded from IP address 59.127.168.43 on 14/10/2019 at 23:50

Searching Method of Structural Similar Subnets in Protein-protein Interaction Networks Based on Quantum Walks

Li-Ping Yang[1,2], Song-Feng Lu[2,3,*]

[1]College of Informatics, Huazhong Agricultural University, Wuhan 430070, China
[2]School of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[3]Shenzhen Research Institute, Huazhong University of Science and Technology, Shenzhen 518063, China

*Corresponding author e-mail: lusongfeng@hotmail.com

Abstract: Comparing and investigation into different protein-protein interaction networks (PPI networks) is significant for discovering new biological function and comprehending the evolution of the protein-protein interactions. Because general PPI networks are large in scale, existing classical computation algorithms of solving alignment and search in PPI networks possess too high time complexity. The time complexity is so high that it is impossible for the algorithm to align the whole network simultaneously. An effective quantum algorithm, Quantum-walks Algorithm for PPI-network Similar Subnets Searching (QPSS), is introduced to improve the situation mentioned above based on the continuous-time quantum-walks model in quantum computation. The process in detail of the QPSS algorithm is demonstrated. Moreover, we discuss the performance evaluation of this algorithm. After the time complexity of QPSS algorithm compares with its classical counterpart, it has been proved that the QMSM obtains a nearly quadratic speed-up.

1. Introduction

Research has shown that almost all proteins cannot function alone, instead, they interact with other proteins to perform their functions. In comparison with the interaction networks of different proteins, it can search the retained area, discover new biological functions, and understand the evolution of protein interaction relationships. In recent years, biomolecular network search methods have attracted more and more attention. Some research groups have conducted research in this area and proposed various algorithms, such as MNAligner [1], NetMatch based on the picture-matching Cytoscape plug-in [2].

However, in general, these algorithms have the disadvantage of too much time complexity. For example, for the MNAligner algorithm, if the quantity of proteins in the searched PPI network is N, the method's time complexity reaches $O(N^4)$. However, the PPI network of common species is very large. For example, there are 4,943 proteins in the PPI network of yeast and 18,440 interactions. Therefore, if we apply the MNAligner algorithm to search such a PPI network, the computational complexity of the algorithm would reach the order of 10^{12}.

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.
Published under licence by IOP Publishing Ltd

Quantum computing provides a new solution to solve the search problem of similar subnets in large-scale PPI networks. For the same problem, compared with traditional computational algorithms, with appropriately designed quantum algorithm for quantum computing, the computational complexity can often be significantly improved. The Shor algorithm [3] and the Grover algorithm are typical examples to represent the superiority of quantum computing. The former proposes a large prime factorization algorithm with a polynomial complexity, while the corresponding traditional method's time complexity is exponential. The latter proposes a search algorithm for unsorted database, the computational complexity of which is $O(N^{1/2})$, while the computational complexity of the traditional algorithm is $O(N)$.

2. Preliminary

2.1. Problem description and classical algorithm for solving the problem

A similar subnet search of a PPI network refers to that under the background of a subnet in a PPI network of a certain species (hereinafter referred to as "the species") as a target subnet, to search the subnet with the most similar structure and biological meaning in the PPI network of another species (hereinafter referred to as "heterogenous species"). This is essentially a problem of network comparison. The comparison of two PPI networks is the comparison of two undirected graphs. The formal definition of it is given below.

Given two PPI networks which are represented by an undirected graph $G_1 = (V_1, E_1)$ and another undirected graph $G_2 = (V_2, E_2)$, respectively. W is defined as a mapping from V_1 to V_2, i.e. $W \subseteq V_1 \times V_2$ .Comparison between the undirected graph G_2 and G_1 corresponds to the mapping W* of the aggregate from V_1 to V_2, the following requirement should be met:

$$sim(G_1, G_2) = \arg\max \sum_{<a,b>\in W^*} sim(a,b) \tag{1}$$

The mapping that maximizes the value of the similarity function between graph G_1 and graph G_2, which is the result of comparison between the PPI networks G_1 and G_2. In Equation (1), $sim(a,b)$ refers to the similarity between the protein node a in the PPI network G_1 and the protein node b in the network G_2. The most common definition of similarity between proteins is the sequence similarity between proteins. Proteins with similar sequences are generally referred to as homologous proteins. The Blast tool is applied to compare the sequences of proteins a and b, and homologous coefficients between proteins a and b can be obtained.

At present, many traditional graph-based algorithms have been proposed to solve the similar subnet search problems of PPI networks, among which the MNAligner method [1] proposed by Li et al. is well-known. The basic steps of the MNAligner algorithm are as follows:

(1) Firstly, obtain the biological network to be examined, such as the adjacency matrix of the PPI network and the similarity matrix between the vertices of the two networks. And the matching matrix X of the two networks is the target to be solved in this problem. The matching matrix X is expressed as $X = \{x_{ij}\}$.

(2) Obtain the adjacency matrix, similarity matrix according to step (1), and the maximum match of the comparison between the two networks is reduced to the target function shown as in the following equation (2). The constraint is a matching rule for the network comparison.

$$\max_X f(G_1, G_2) = \lambda \sum_{i=1}^{m}\sum_{j=1}^{n} s_{ij}x_{ij} + (1-\lambda)\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{k=1}^{m}\sum_{l=1}^{n} a_{ik}b_{jl}x_{ij}x_{kl} \tag{2}$$

(3) Through modelling following steps (1) and (2), the network comparison problem is reduced as an integer quadratic programming problem through further relaxing the constraint, then apply the existing method to solve it.

It is not difficult to conclude from equation (2) that the computational complexity of the MNAligner algorithm is higher; assuming that the larger of the protein quantity of the PPI networks G_1 and G_2 is $N_{\max}$, so, this method's time complexity is $O(N_{\max}^4)$.

2.2. Overview of Continuous Time Quantum Walks

The process of continuous quantum walk on a graph *G = (V, E)* is detailed as follows. The nodes set *V* is the whole state space for the quantum random walk. The quantum walk migration process is only performed between adjacent nodes, and the probability of migration is equal to the reciprocal of the degree d(*u*) of the current node u, i.e. 1/*d*(*u*).

At any time t, the state of the continuous quantum walk on graph G is a superposition of all ground states, represented by a state vector $|\varphi_t\rangle$:

$$|\varphi_t\rangle = \sum_{u \in V} \alpha_u(t)|u\rangle \tag{3}$$

A unitary transformation can realize the migration process of continuous quantum walk. The equation (4) as follows introduces the time-dependent evolution process of the state vector $|\varphi_t\rangle$.

$$\frac{d}{dt}|\varphi_t\rangle = -iL|\varphi_t\rangle \tag{4}$$

In the equation (4), *L*=D-A, and *L* is graph G's Laplacian matrix, D is G's diagonal degree matrix, A is G's adjacency matrix. Given initial state of a continuous quantum walk, i.e. the initial value $|\varphi_t\rangle$ of a given state vector $|\varphi_0\rangle$, it is possible to obtain the state vector $|\varphi_t\rangle$ at any time t according to Equation (4), which is shown as in Equation (5), thereby, the probability distribution of quantum walking in each ground state is obtained.

$$|\varphi_t\rangle = e^{-iLt}|\varphi_0\rangle \tag{5}$$

2.3. Non-isomorphic discrimination algorithm based on continuous quantum walks

In view of the many drawbacks of the traditional computational algorithms described in Section 2.1, we propose a quantum algorithm QPSS based on continuous quantum walk that integrates the Qiang's graph isomorphism based on quantum walk [4]. The following is a brief introduction to Qiang's non-homomorphic discrimination algorithm based on continuous quantum walk (herein referred to as the "QUID algorithm").

Qiang's distinction non-homomorphic graphs method[4] refers to judge whether the two graphs are isomorphic by the algorithm for the given G_1, G_2 graphs. The main steps of the QUID algorithm are as follows: 1) Select one node set V1 in the graph G_1 arbitrarily, then increase *Dmax* self-loops for all the nodes. 2) Starting from the initial state $|\varphi_0\rangle$, continuous quantum walk is performed on the graph G_1. 3) At time T, we meter the set V_1's probability amplitudes. 4) Repeat steps 1)-3) until we obtain G_1's all node sets' probability amplitudes. 5) Perform identical above steps 1)-4) on G_2. 6) Calculate the respective individual ID of G_1 and G_2, then we can obtain the mixed ID of G_1 and G_2. 7) We calculate individual ID and mixed ID for multiple times via repeating steps 1)-6). 8) Compare G_1 and G_2's individual IDs: if (we find that at some time they are not equal) the two graphs are not isomorphic and the whole algorithm finishes; else continue to compare both individual IDs at the next time. 9) We compare G_1 and G_2's mixed ID: if (we find that at some time they are not equal) the two figures are not isomorphic; else if their mixed IDs are always equal, the two graphs are isomorphic and the algorithm finishes.

2.4. Isomorphic mapping search algorithm based on comparing probability amplitudes

In our quantum algorithm QPSS, Qiang's homogeneous mapping search algorithm[4] based on probability amplitude comparison (herein referred to as "PIMS algorithm") is also applied.

The main steps of the PIMS algorithm are as follows: 1) arbitrarily select the node u in the graph G and add Dmax self-loops. 2) Starting from the initial state $|\varphi_0\rangle$, continuous quantum walking is performed on the graph G. 3) At time T, we measure the node *u*'s probability amplitude and denote it

by $\alpha_u(T)$. 4) Perform steps 1)-3) by sequence on all the nodes in the graph G, and a set W composed of N probability amplitudes is obtained, which is denoted as $\alpha_G(T)$. 5) Perform the above steps (1)-(4) on the graph H to obtain a set of probability amplitudes $\alpha_H(T)$. 6) Difference is made between each two elements in $\alpha_G(T)$ and $\alpha_H(T)$, and the probability amplitude difference matrix is obtained. 7) Find out the elements with N values of 0 in the probability amplitude difference matrix, and obtain the isomorphic mapping relationship between graphs G and H according to their row and column coordinates.

3. A novel solution based on quantum walks to the above-mentioned problem

3.1. Basic idea of the quantum algorithm QPSS

For the search problem of similar subnets in the PPI network, the target is to apply a subnet in the PPI network of the species as the target subnet. In the PPI network of the heterogeneous species, to search the subnet that is the most similar in topology and biological significance to the target subnet.

The basic idea of the proposed quantum algorithm QPSS is to first select a target subnet G1 in the PPI network of the species, and construct a queue for all the nodes in G_1 according to the descending order of the degrees of the nodes. For each protein node a_i in G_1 (its degree recorded as d_i), search for all nodes with a degree not less than d_i, and construct a queue of nodes in G_2 on ai accordingly in the heterogeneous PPI network G_2 to be searched. Next, construct a linked list L_{ij} for the sub-network G_1 to be detected in G_2. Then, for the graphs G_1 and H_2, the quantum algorithm QUID described in section 2.3 is applied to determine whether both are isomorphic. If both are non-isomorphic, it will search for the subnet H_2 to be detected again. If both are isomorphic, it continues to apply the algorithm PIMS as described in Section 2.4 to search for matching pairs of proteins in G_1 and H_2. The entire search process ends until the matching of all proteins in the two networks is completed.

Table 1 Steps of the algorithm QPSS

(1) For the target subnet G_1, a queue is constructed according to the descending order of each node in G_1.
(2) For each protein node a_i (i=1,...,n) in G_1, denote its degree by d_i; in the PPI network G_2 to be searched, construct a queue about a_i for all nodes with a degree not less than d_i : $Q_i=\{b_{i1}, b_{i2}, ..., b_{im}\}$.
(3) Construct a queue L_{ij} for the subnet H_2 to be searched in G_2: firstly, place b_{ij} (initial i=1, j=1) into the queue L_{ij}.
(4) The adjacency point (denoted by b_{i+1}) of the node b_i (initially i=1) in G_2 is placed in the queue L_{ij}. If i=n-1, it indicates that the subnet H_2 to be searched has been all generated, and the process proceeds to step (5); otherwise, i+1 is assigned to i, and the process proceeds to step (4).
(5) For the graph $G_1=(V_1 , E_1)$ and $H_2=(V_2 , E_2)$, where $V_1 = \{ a_1,a_2,\ldots\ldots,a_n \}$, $V_2 = \{ b_1,b_2,\ldots\ldots,b_n \}$, apply the above-mentioned sub-algorithm QUID to determine whether the graphs G_1 and H_2 are isomorphic. If both are isomorphic, go to step (7).
(6) When G_1 and H_2 are non-isomorphic, first increase the value of j by one; if the value of j does not exceed n, then the value of i is unchanged, and the process proceeds to step (3). If the value of j exceeds n, then the value of i is increased by one; and then, if the value of i exceeds n, the algorithm ends with failure; otherwise, the value of j is set to 1, then the process proceeds to step (3).
(7) Using the sub-algorithm PIMS, the isomorphic mapping relationship *f* of all node pairs between the graphs G_1 and H_2 is obtained.
(8) Construct a similarity matrix S about PPI networks G_1 and H_2.
(9) Verify whether there is similarity matching relationship in the biological meaning between nodes a_i and $b_{isoGH(i)}$ which has been confirmed to be an isomorphic mapping node pair between G_1 and H_2 in step (7). Among all the node pairs with isomorphic mapping relationship, if all the similar matching protein node pairs are searched, the process proceeds to step (10).
(10) Among the remaining nodes in G_1 and H_2, apply the NBM algorithm repeatedly, until all the similar matching protein pairs in G_1 and H_2 are found.

3.2. Steps of the algorithm QMSM

Table 1 is the flow of steps of the QPSS algorithm, among which, G_1 represents the target subnet in the PPI network of the species, and H_2 represents the structurally similar subnet to be searched in the PPI network of the heterogeneous species.

The relevant details of the algorithm shown in Table 1 are as follows:

In step (2), for each protein node a_i in G_1 (the degree is d_i), all nodes in G_2 with a degree not less than d_i are searched, and construct a queue Q_i according to the descending order of homologous

coefficient values with a_i (obtained by Blast).

In step (4), the adjacency point b_{i+1} of the protein node b_i in the network G_2 is the protein node having an interaction relationship with the protein node b_i. It is possible that there are many such candidate nodes in G_2, and the node with the highest homologous coefficient value with the node $a_{(i+1) \bmod n}$ is taken into the queue L_{ij} as b_{i+1}.

In step (9), whether there is a similarity matching relationship between the protein node a_i and the $b_{isoGH(i)}$ is verified by means of testing whether the corresponding value $S_{i,isoGH(i)}$ of a_i and $b_{isoGH(i)}$ in the similarity matrix is not less than the empirical value ε. That is, if the $S_{i,isoGH(i)}$ meets Equation (6), the protein nodes a_i and $b_{isoGH(i)}$ are matched with similarity. Generally, the empirical value is taken as $\varepsilon \geq 0.8$.

$$S_{i,isoGH(i)} \geq \varepsilon \quad (i \in V_1,\ \mathrm{isoGH}(\ i\ ) \in V_2) \tag{6}$$

4. Performance analysis of QPSS

It is not difficult to conclude that the computational complexity of the quantum algorithm QPSS is mainly determined by the complexity of the sub-algorithms QUID and PIMS.

For the sub-algorithm QUID, if the size of the node set is K and the quantity of protein nodes of the target sub-network G_1 is N, there are a total of NK different node sets. Therefore, it is necessary to perform NK continuous quantum walks for the probability amplitude of all node sets according to the QUID algorithm. According to the computational complexity of the simulation of a continuous quantum walk on a traditional computer, the computational complexity required for N^K consecutive quantum walks is $O(N^K \times N^3) = O(N^{K+3})$. There are also a total of N^K probability amplitude sets of N^K node sets, and the computational complexity to get the logarithm of similar node sets requires a computational complexity of $O(N^K \times N^K) = O(N^{2K})$. So, if K=1, the total time complexity of QUID is $O(N^4)$. The Grover technology can achieve acceleration of the continuous quantum walk simulation, reducing the computational complexity of continuous quantum walk simulation from $O(N^3)$ to $O(N^{3/2})$. Thus, when K=1, the complexity of the sub-algorithm QUID is reduced to $O(N^{5/2})$. In the PIMS part of the sub-algorithm, under the circumstance where the number of nodes of the target sub-network G_1 is N, the computational complexity of implementing simulation of consecutive quantum walks by N times is $O(N \times N^3) = O(N^4)$. The computational complexity when constructing the probability amplitude difference matrix is $O(N^2)$. So, PIMS's total time complexity is $O(N^4 + N^2) = O(N^4)$. Similarly, the Grover technology can be applied to accelerate the continuous quantum walk simulation; thus, the complexity of the sub-algorithm PIMS can be reduced to $O(N^{5/2})$.

In conclusion, the computational complexity of QPSS algorithm is $O(N^{5/2} + N^{5/2}) = O(2N^{5/2})$. Compared with the traditional computer algorithm for solving the same problem as described in Section 2.1, the computational complexity almost achieves secondary acceleration (for example, the method MNAligner's time complexity is about $O(N^4)$). Especially for the PPI network comparison problem, since the network size is usually large, the time complexity of the traditional calculation

algorithm reaches an order of magnitude more than 10^{12}; furthermore, because the time complexity is too high, it is often difficult for the traditional algorithm to achieve the comparison of the entire network at one time. Therefore, the quantum algorithm QPSS shows a clear advantage in solving this problem.

5. Conclusions

In this paper, the search problem of similar subnets in PPI networks is studied. Firstly, the disadvantages of the existing traditional computational algorithms are analyzed, and the quantum algorithm QPSS based on quantum walking is thus proposed. Based on integration of non-homogeneous discrimination quantum algorithm and the homogeneous mapping search algorithm on the basis of probability amplitude comparison, combined the biological meaning in the PPI network comparison, an algorithm QPSS for solving similar subnet searches in PPI networks is proposed as an improvement. The execution steps of the QPSS algorithm are described and the performance of the algorithm is analyzed. Compared with the previous traditional algorithm, the analysis proves that the quantum algorithm QPSS achieves secondary acceleration in terms of time complexity and other performance.

References

[1] Li Zhenping，Zhang Shihua，Wang Yong，et al. Alignment of molecular networks by integer quadratic programming [J].Bioinformatics. 2007，23: 1631-1639

[2] Ferro A，Giugno R，Pigola G, et al. Net match-A Cytoscape plugin for searching biological networks[J].Bioinformatics. 2007, 23(7)：910-912

[3] Shor, P.W. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer [J]. SIAM review, 1999, 41(2): 303-332.

[4] Qiang Xiaogang, Yang Xuejun, Wu Junjie, et al. An enhanced classical approach to graph isomorphism using continuous-time quantum walk. J. Phys. A: Math. Theor. 45 (2012) 045305