frontiers in Genetics

ORIGINAL RESEARCH
published: 29 July 2021
doi: 10.3389/fgene.2021.721486

A Novel Model for Identifying Essential Proteins Based on Key Target Convergence Sets

Jiaxin Peng[1,2], Linai Kuang[1], Zhen Zhang[2], Yihong Tan[2], Zhiping Chen[2] and Lei Wang[1,2]**

[1] College of Computer, Xiangtan University, Xiangtan, China, [2] College of Computer Engineering and Applied Mathematics, Changsha University, Changsha, China

In recent years, many computational models have been designed to detect essential proteins based on protein-protein interaction (PPI) networks. However, due to the incompleteness of PPI networks, the prediction accuracy of these models is still not satisfactory. In this manuscript, a novel key target convergence sets based prediction model (KTCSPM) is proposed to identify essential proteins. In KTCSPM, a weighted PPI network and a weighted (Domain-Domain Interaction) network are constructed first based on known PPIs and PDIs downloaded from benchmark databases. And then, by integrating these two kinds of networks, a novel weighted PDI network is built. Next, through assigning a unique key target convergence set (KTCS) for each node in the weighted PDI network, an improved method based on the random walk with restart is designed to identify essential proteins. Finally, in order to evaluate the predictive effects of KTCSPM, it is compared with 12 competitive state-of-the-art models, and experimental results show that KTCSPM can achieve better prediction accuracy. Considering the satisfactory predictive performance achieved by KTCSPM, it indicates that KTCSPM might be a good supplement to the future research on prediction of essential proteins.

Keywords: protein-protein interaction, essential protein, heterogeneous network, random walk with restart, key target convergence set

Edited by:
Quan Zou,
University of Electronic Science and Technology of China, China

Reviewed by:
Yuhua Yao,
Hainan Normal University, China
Xing Chen,
China University of Mining and Technology, China
Bing Wang,
Anhui University of Technology, China

***Correspondence:**
Linai Kuang
kla@xtu.edu.cn
Lei Wang
wanglei@xtu.edu.cn

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *07 June 2021*
Accepted: *30 June 2021*
Published: *29 July 2021*

Citation:
Peng J, Kuang L, Zhang Z, Tan Y, Chen Z and Wang L (2021) A Novel Model for Identifying Essential Proteins Based on Key Target Convergence Sets. Front. Genet. 12:721486. doi: 10.3389/fgene.2021.721486

INTRODUCTION

With the deepening of researches on proteins, accumulating evidences have demonstrated that proteins are closely related to most of the life activities. Moreover, different proteins are of different importance to different life activities. Among these proteins, essential proteins, as a kind of important proteins, are essential for the survival, and development of life. Therefore, in recent years, detection and recognition of essential proteins has become a hot issue in the research and development of disease treatment. However, it is very time-consuming and expensive to identify essential proteins by traditional biological experiments, which leads to the emergence and development of computational prediction methods. For instance, Zhao et al. (2019) designed a new random walk wandering based prediction model to detect key proteins based on a heterogeneous network consisting of proteins and protein domains. Jeong et al. (2001) found that PPI networks are scale-free and proposed a center-lethal rule for PPI networks. Based on which, lots of methods including the information centrality (IC) (Stephenson and Zelen, 1989), betweenness centrality (BC) (Joy et al., 2014), degree centrality (DC) (Hahn and Kern, 2004), Closeness Centrality (CC) (Wuchty and Stadler, 2003), subgraph centrality (SC) (Estrada and Rodriguez-Velazquez, 2005), and

neighbor centrality (NC) (Wang et al., 2012) had been put forward successively. In addition, Yu et al. (2007) proposed the importance of network bottlenecks. Estrada (2006) found that a small number of binary proteins were mostly essential proteins. Chua et al. (2006) proposed to identify essential proteins by calculating the weights of indirect neighbor nodes. Li et al. (2012) designed a method to predict essential proteins by combining PPI networks with gene expression data of proteins. Li et al. (2011) find essential proteins by analyzing the relationship between proteins and their neighbors, and define the method as LAC. Peng et al. (2012) combined orthology information of proteins with PPI networks to predict key proteins. Zhao et al. (2014) found that combination of gene expression profiles and PPI networks was of great help to the prediction accuracy of essential proteins. Min et al. (2017) discovered that the complex information of proteins can improve prediction accuracy and precision of potential essential proteins. Zhao et al. (2016) proposed a basic protein identification method based on protein gene time expressions and protein domains. Zhang et al. (2019) proposed a new protein prediction method called TEGS, which can identify essential proteins by fusing the introduced multiple biological information data. Lei et al. (2018) found that it can achieve good results to adopt artificial fish swarm optimization algorithm into key protein prediction. Peng et al. (2015) discovered that combination of protein domain features and protein interaction networks can effectively predict potential essential proteins. Li et al. (2019) proposed a target convergence set (TCS) based method for predicting potential lncRNA-disease associations. Athira and Gopakumar (2020) proposed a multiplex network to identifying essential proteins. Zhang et al. (2018) designed a novel method by combining network topology, gene expression profile and GO information to identifying essential proteins. Fan et al. (2017) proposed a modified PageRank algorithm based on subcellular information. Meng et al. (2021) predict the essential protein by constructing a new weighted protein and protein domain network, and performing a local random walk on this basis. Xenarios et al. (2002) introduced a public database called DIP for studying cellular networks of protein interactions. Gavin et al. (2006) provided a complete and comprehensive eukaryotic machine and biological data integration and modeling platform.

Inspired by above methods, in this manuscript, a computational model named KTCSPM was proposed to predict essential proteins. In KTCSPM, a weighted PDI network was first constructed by integrating a weighted PPI network and a weighted domain-domain interaction (DDI) network. And then, each node in the weighted PDI network would be assigned a unique key target convergence sets (KTCS) according to the network distance information of the weighted PDI network, which could reflect the specificity of different nodes in the process of random walk with restart and improve the predictive performance of KTCSPM. Next, for an arbitrarily selected walker, considering that there may still be some nodes that are essential proteins but not included in KTCS while KTCS reached the final convergence state, each node in the heterogeneous network would be further assigned a unique Intact Set (IS) to ensure that the predicted results would not be omitted as far as possible. Next, we will construct a random walk probability matrix and calculate the stable walk probability of all nodes, and then rank each protein based on the initial protein score vector. Finally, in order to evaluate the predictive performance of KTCSPM, we compared it with 12 advanced predictive methods based on two kinds of yeast PPI networks, and experimental results showed that KTCSPM can achieve reliable predictive accuracy of 90.19, 81.96, 70.72, 62.04, 55.83, and 51.13% in top 1, top 5, top 10, top 15, top 20, and top 25% of predicted key proteins separately, which are better than all these 12 competing predictive models.

MATERIALS AND METHODS

Construction of the Weighted PPI Network

In this section, we will download known PPI data from two different public databases such as the DIP database (Xenarios et al., 2002) and the Gavin database (Gavin et al., 2006), respectively. Obviously, based on these known PPI network downloaded from any given public database, an original PPI network $\boldsymbol{PPIN} = < \boldsymbol{D_{PP}}, \boldsymbol{E_{PP}} >$ can be constructed as follows: Let $\boldsymbol{D_{PP}} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \boldsymbol{p_{N_P}}\}$ represent the set of newly downloaded proteins and $\boldsymbol{E_{PP}}$ denote the set of edges between proteins in PPIN, here, for any two given proteins $\boldsymbol{p_i}$ and $\mathbf{p_j}$ in $\boldsymbol{D_{PP}}$, if and only if there is a known interaction between them, then we define that there is an edge between them in PPIN. Thereafter, based on the newly constructed original PPI network PPIN, we can further obtain an $\boldsymbol{N_P} \times \boldsymbol{N_P}$ dimensional adjacency matrix $\boldsymbol{M_{PPIN}}$ as follows: for any two given protein nodes $\boldsymbol{p_i}$ and $\boldsymbol{p_j}$ in PPIN, if and only if there is an edge between them in PPIN, there is $\boldsymbol{M_{PPIN}}(\boldsymbol{p_i}, \mathbf{p_j}) = 1$, otherwise there is $\boldsymbol{M_{PPIN}}(\boldsymbol{p_i}, \mathbf{p_j}) = 0$.

In previous studies, the Gaussian interaction profile kernel similarity has been widely used to measure the similarity between similar nodes (Chen et al., 2016). In this section, for any two given proteins p_i and p_j in M_{PPIN}, we define the Gaussian interaction profile kernel similarity between them as follows:

$$\boldsymbol{GKS}(\boldsymbol{i}, \boldsymbol{j}) = \boldsymbol{exp}\left(-\boldsymbol{\gamma_p}||\boldsymbol{IP}(\boldsymbol{p_i}) - \boldsymbol{IP}(\boldsymbol{p_j})||^2\right) \tag{1}$$

$$\boldsymbol{\gamma_p} = \boldsymbol{\gamma'_p} \Big/ \sum_{\boldsymbol{k}}^{\boldsymbol{N_P}} = \boldsymbol{1}||\boldsymbol{IP}(\boldsymbol{p_k})||^2 \tag{2}$$

Here, $IP(p_t)$ represents the vector of elements in the *t-th* row of the matrix M_{PPIN}, and γ_p denotes the normalized kernel bandwidth based on the bandwidth parameter γ'_p. In addition, according to the methodology proposed by Vanunu et al. (2010), we will further optimize above Gaussian interaction profile kernel similarity of protein by introducing a logistics function as follows:

$$LGKS(p_i, p_j) = \frac{1}{1 + e^{(-12GKS(i,j) + log9999)}} \tag{3}$$

This logistic function can make the calculated results of Gaussian interaction porofile kernel similarity more influential in the identification of essential proteins. Additionally, considering that while analyzing the topology structure of PPI network, the

PPI network can be weighted to show the interaction between proteins, therefore, based on above newly obtained matrixLGKS, for any two given proteins p_i and p_j, we can weigh the relationship between them as follows:

$$W_{PP}(p_i, p_j) = \frac{LGKS(p_i, p_j) + \frac{|N(p_i) \cap N(p_j)|^2}{(|N(p_i)+1|)*(|N(p_j)+1|)}}{2} \quad (4)$$

Here, $N(p_i)$ and $N(p_j)$ represent the sets of protein nodes directly adjacent to p_i and p_j in PPIN, respectively, and $N(p_i) \cap N(p_j)$ denotes the set of protein nodes adjacent to both p_i and p_j in PPIN. Obviously, based on above Equation (4), we can obtain a weighted PPI network $WPIN = < D_{PP}, E_{WPP} >$ easily by taking $W_{PP}(p_i, p_j)$ as the weight of the edge between nodes p_i and p_j in WPIN, where D_{PP} and E_{WPP} denote the sets of nodes and edges in WPIN separately.

Construction of the Weighted DDI Network

In this section, we will first download known domain data from the Pfam database (Peng et al., 2012; Bateman et al., 2014), and for convenience, let $\boldsymbol{D_{DD}} = \{\boldsymbol{d_1}, \boldsymbol{d_2}, \ldots, \boldsymbol{d_{N_D}}\}$ represent the set of newly downloaded domains, then for any given protein $\boldsymbol{p_i} \in \boldsymbol{D_{PP}}$, and domain $\boldsymbol{d_j} \in \boldsymbol{D_{DD}}$, it is obvious that we can estimate the relationship between them as follows:

$$\boldsymbol{W_{PD}(p_i, d_j)} = \frac{\sum_{p_k \in d_j} \boldsymbol{W_{PP}(p_i, p_k)}}{|\boldsymbol{d_j}|} \quad (5)$$

Here, $|\boldsymbol{d_j}|$ represents the number of different proteins belonging to $\boldsymbol{d_j}$. Furthermore, according to above Equation (5), for any two given domains $\boldsymbol{d_i}$ and $\boldsymbol{d_j}$ in $\boldsymbol{D_{DD}}$, we can calculate the relationship between them as follows:

$$\boldsymbol{W_{DD}(d_i, d_j)} = \frac{\sum_{p_x \in d_i} \boldsymbol{W_{PD(p_x, d_j)}} + \sum_{p_y \in d_j} \boldsymbol{W_{PD}(p_y, d_i)}}{|\boldsymbol{d_i}| + |\boldsymbol{d_j}|} \quad (6)$$

Obviously, based on above Equation (6), we can easily construct a weighted DDI network $WDIN = < \boldsymbol{D_{DD}}, \boldsymbol{E_{DD}} >$ as follows: Let $\boldsymbol{E_{DD}}$ denote the set of edges between domains in WDIN, here, for any two given domains $\boldsymbol{d_i}$ and $\mathbf{d_j}$ in $\boldsymbol{D_{DD}}$, if and only if there is $\boldsymbol{W_{DD}(d_i, d_j)} > 0$, we define that there is an edge between them in WDIN, and at the same time, the weight of the edge between $\boldsymbol{d_i}$ and $\mathbf{d_j}$ is $\boldsymbol{W_{DD}(d_i, d_j)}$.

Construction of the Weighted PDI Network

Based on above Equations (4)–(6), it is obvious that we can construct a new $\boldsymbol{(N_P + N_D) \times (N_D + N_P)}$ dimensional matrix $\boldsymbol{M_{PD}}$ as follows:

$$\boldsymbol{M_{PD}} = \begin{bmatrix} \boldsymbol{W_{PP}} & \boldsymbol{W_{PD}} \\ \boldsymbol{W_{PD}^T} & \boldsymbol{W_{DD}} \end{bmatrix} \quad (7)$$

Here, $\boldsymbol{W_{PD}^T}$ is a transport matrix of $\boldsymbol{W_{PD}}$. Based on above matrix $\boldsymbol{M_{PD}}$, we can easily construct a novel weighted PDI network $WPDIN = < \boldsymbol{D_{PD}}, \boldsymbol{E_{WPD}} >$ as follows: Let $\boldsymbol{D_{PD}} = \{pd_1, pd_2, \ldots, \boldsymbol{pd_{N_P}}, \boldsymbol{pd_{N_P+1}}, \boldsymbol{pd_{N_P2}}, \ldots, \boldsymbol{pd_{N_P}} + \boldsymbol{N_D}\} = \{\boldsymbol{p_1}, \boldsymbol{p_2}, \ldots, \mathbf{p_{N_P}}, \boldsymbol{d_1}, \boldsymbol{d_2}, \ldots, \boldsymbol{d_{N_D}}\}$ represent the set of nodes in WPDIN, and $\boldsymbol{E_{WPD}}$ denote the set of edges in WPDIN, then, for any two given nodes $\boldsymbol{pd_i}$ and $\boldsymbol{pd_j}$ in $\boldsymbol{D_{PD}}$, if and only if there is $\boldsymbol{W_{PP}}(pd_i, pd_j) > 0$ or $\boldsymbol{W_{PD}(pd_i, pd_j)} > 0$ or $\boldsymbol{W_{DD}(pd_i, pd_j)} > 0$, there is an edge between them in $\boldsymbol{E_{WPD}}$, and moreover, the weight of the edge between them is $\boldsymbol{M_{PD}(pd_i, pd_j)}$.

Calculation of Initial Scores for Proteins

For any given protein node pi in WPDIN, in this section, we will assign an initial score for it based on the functional features extracted from the subcellular localization information of proteins, and the conservative features provided by orthologous information of proteins. Firstly, we will download the orthologous information of proteins from the InParanoid database (Mewes et al., 2006; Gabriel et al., 2010) and the subcellular localization information of proteins from the COMPART-MENTS database (Binder et al., 2014; Min et al., 2017). And then, for convenience, let $\boldsymbol{Np(i)}$ represent the total number of proteins relating to the *i-th* subcellular localization, $\boldsymbol{N_L}$ denote the total number of different subcellular localizations downloaded above, and $\boldsymbol{S(p_i)}$ represent the set of subcellular locations associating with $\boldsymbol{p_i}$. Hence, we can calculate a score for $\boldsymbol{p_i}$ based on the subcellular localization information as follows:

$$\boldsymbol{Subcell\ \_\ Score\ (p_i)} = \mathbf{max}_{\boldsymbol{j \in S(p_i)}} \boldsymbol{Subcell(j)} \quad (8)$$

Where,

$$\boldsymbol{Subcell\ (j)} = \frac{\boldsymbol{N_p(j)}}{\mathbf{max}_{\mathbf{1 \le k \le N_L}}(N_p(k))} \quad (9)$$

Next, let $Hom(p_i)$ denote the score of p_i in the downloaded homologous information and N_H denote the total number of proteins with homologous information, then, we can calculate another score for p_i based on the homologous information as follows:

$$Hom\_Score\ (p_i) = \frac{Hom(p_i)}{\max_{1 \le j \le N_H} Hom(p_i)} \quad (10)$$

Finally, through integrating above two kinds of scores together, we can obtain an initial score for p_i as follows:

$$Initial\_Score\ (p_i) = \frac{Subcell\_Score(p_i) + Hom\_Score(p_i)}{2} \quad (11)$$

Construction of the Prediction Model KTCSPM

Establishment of the Key Target Convergence Sets

Before implementing random walk with restart on WPDIN, as shown in **Figure 1**, each node in WPDIN will establish a unique KTCS first according to the following steps:

Step 1: For any given protein node p_i in WPDIN, we define its original KTCS as the set of all domain nodes associating with p_i, that is the original KTCS of p_i is $KTCS_0(\mathrm{p_i}) = \{d_k \mid M_{PD}(d_k, p_i) = 1, d_k \in D_{DD}\}$. Similarly, for any given protein domain node d_j, we can define its original KTCS as $KTCS_0(d_j) = \{p_k \mid M_{PD}(d_j, p_k) = 1, p_k \in D_{PP}\}$.

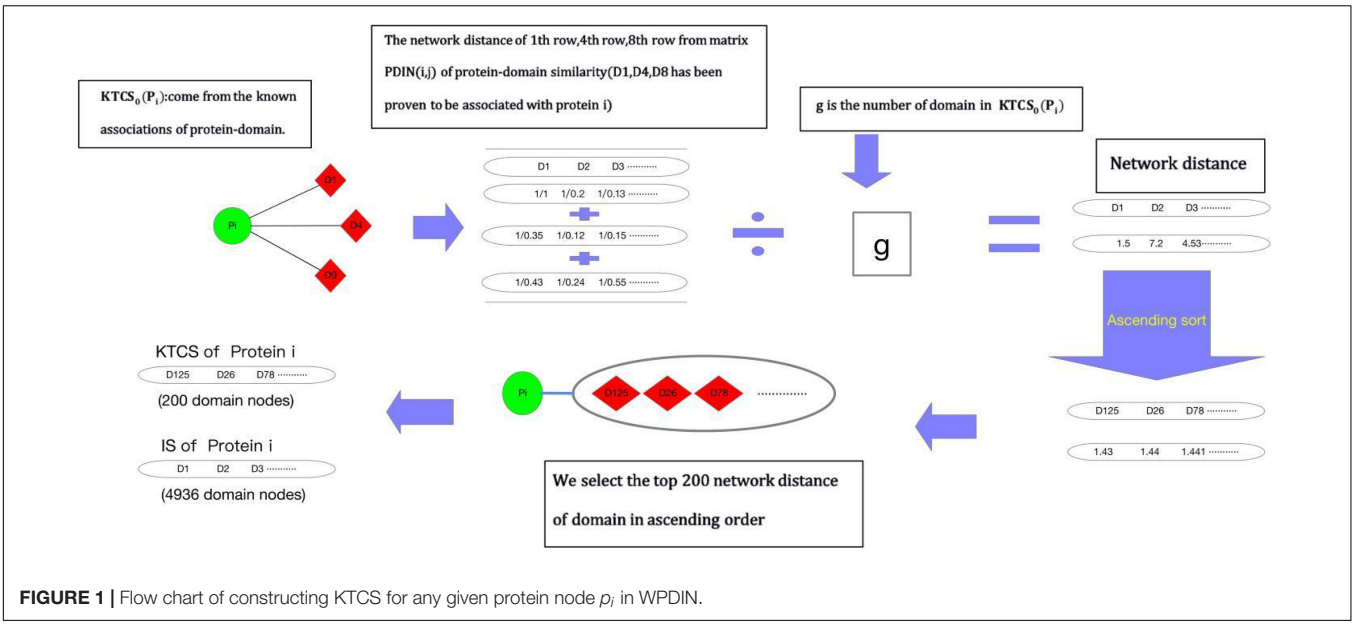

FIGURE 1 | Flow chart of constructing KTCS for any given protein node p_i in WPDIN.

Step 2: For any given protein node p_i in WPDIN, $\forall d_k \in KTCS_0(p_i)$ *and* $\forall d_t \in D_{DD}$, we define the network distance between d_k and d_t in WPDI as follows:

$$AD(d_k, d_t) = \frac{1}{W_{DD}(d_k, d_t)} \tag{12}$$

Similarly, for any given domain node d_i in WPDIN, $\forall p_k \in KTCS_0(\mathrm{d_i})$ *and* $\forall p_t \in D_{PP}$, we can define the network distance between p_k and p_t in WPDI as follows:

$$AD(p_k, p_t) = \frac{1}{W_{PP}(p_k, p_t)} \tag{13}$$

Step 3: According to the above Equations (13, 14), for any given protein node p_i or domain node d_j in WPDIN, we define the KTCS (d_j) of d_j as the set of first 200 protein nodes in WPDIN that have the minimum average network distance to nodes in $KTCS_0(d_j)$, and the *KTCS* (p_i) of p_i as the set of first 200 domain nodes in WPDIN that have the minimum average network distance to nodes in $KTCS_0\left(p_i\right)$. Therefore, it easy to know that these 200 protein nodes in *KTCS* (d_j) may belong to $KTCS_0\left(d_j\right)$ or may not belong to $KTCS_0\left(d_j\right)$, and these 200 domain nodes in $KTCS(p_i)$ may belong to $KTCS_0\left(p_i\right)$ or may not belong to $KTCS_0\left(p_i\right)$ as well.

Random Walk With Restart in WPDIN

The transition process of a walker from a starting node in the network to other nodes with a given probability is called the method of Random walk. Based on the assumption that there is a correlation between essential proteins and domains, as shown in **Figure 2**, the random walk process of KTCSPM can be mainly divided into the following steps:

Step 1: For a walker, before it starts to walk randomly in WPDIN, we can first obtain a transition probability matrix W for it as follows:

$$W(i, j) = \frac{M_{PD}(i, j)}{\sum_{k=1}^{N_P+N_D} M_{PD}(i, k)} \tag{14}$$

Step 2: Moreover, for any given node pd_i in WPDIN, we can as well obtain an initial probability vector $R_i(0)$ for the walker as follows:

$$R_i(0) = (R_{i,1}(0), R_{i,2}(0), \ldots R_{i,j}(0), \ldots R_{i,N_P+N_D}(0)) \tag{15}$$

$$R_{i,j}(0) = W\left(i, j\right), j = 1, 2, 3, \ldots N_P + N_D \tag{16}$$

Step 3: Next, while starting a walk, the walker will select a node (for convenience, let it be pd_0) in WPDIN randomly as its initial location of this walk, where pd_0 may be a protein node or a domain node. Supposing that after walking t-1 hops, the walker reaches the current node pd_i in WPDIN, then, we can further calculate a new walking probability vector $R_i(t)$ for it as follow:

$$R_i(t) = (1 - \alpha) * W^T * R_i(t - 1) + \alpha * R_i(0) \tag{17}$$

Here, $\alpha(0 < \alpha < 1)$ is a parameter for adjusting weights between $R_i(0)$ and $R_i(t\text{-}1)$. Moreover, for convenience, let $R_i(t) = (R_{i,1}(t), R_{i,2}(t), \ldots, R_{i,j}(t), \ldots, R_{i,N_PN_D}(t))^T$, where $R_{i,j}(t)$ denotes the walking probability that the walker will walk from its current location pd_i to the node pd_j at its *t-th* hop. Here, it is worth noting that for the starting node pd_0, since the walker can be considered to reach $\mathrm{pd_0}$ from pd_0 after zero hops, therefore, for the starting node pd_0, the walker can obtain an initial probability vector $R_0(0)$, and a walking probability vector $R_0(1)$.

Step 4: Assuming that the walker has walked from a node pd_i to a current node pd_j after t-1 hops during its random walk in

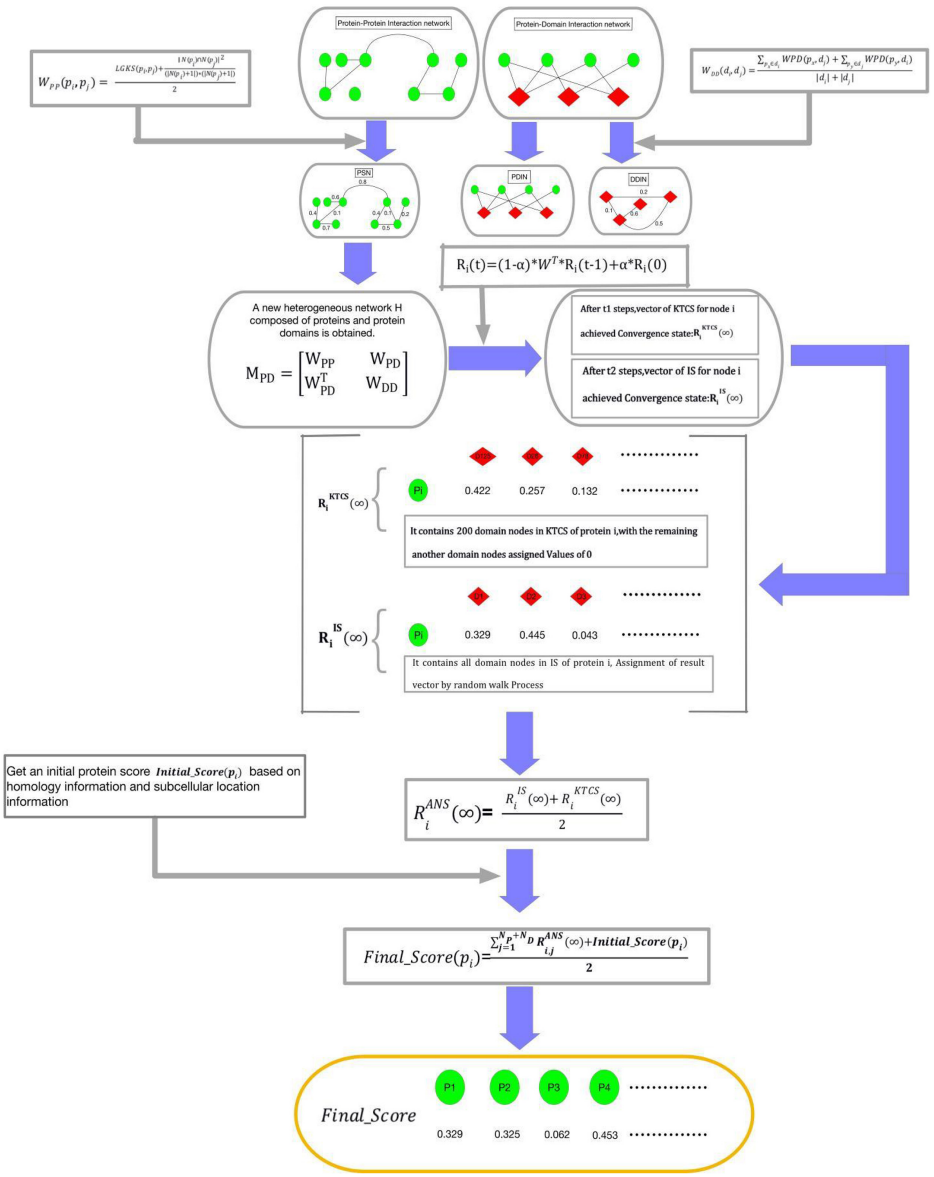

FIGURE 2 | Schematic diagram of the construction process of KTCSPM, where green nodes represent proteins and red nodes represent domains.

WPDIN, the walk probability vectors calculated by the walker at pd_i and pd_j are $R_i(t\text{-}1)$ and $R_j(t)$, respectively, if the L1 norm between $R_i(t\text{-}1)$ and $R_j(t)$ satisfies $||R_j(t) - R_i(t-1)||_1 \leq 10^{-6}$, then we define that the walking probability vector $R_j(t)$ has reached a stable state at its current location. Moreover, after the walker having obtained a stable walking probability at each node in WPDIN, for convenience, we will define the stable probability obtained by the walker at any given node pd_k in WPDIN as $R_k(\infty)$, and then, we can construct a stable walking probability matrix $K(\infty)$ as follows:

$$K(\infty) = \begin{bmatrix} K_1 & K_2 \\ K_3 & K_4 \end{bmatrix} = (R_1(\infty), R_2(\infty), R_3(\infty)....., R_{N_P N_D}(\infty))^T \tag{18}$$

where, K_1 is a $N_P \times N_P$ dimensional matrix, K_2 is a $N_P \times N_D$ dimensional matrix, K_3 is a $N_D \times N_P$ dimensional matrix, and K_4 is a $N_D \times N_D$ dimensional matrix. Thereafter, it is obvious that K_2 and K_3 will be the final result matrices, which can be adopted to predict potential essential proteins.

According to above steps of KTCSPM, it is easy to see that, for any node pd_i in WPDIN, a stable walking probability vector $R_i(\infty) = (R_{i,1}(\infty), R_{i,2}(\infty), \ldots, R_{i,j}(\infty), \ldots, R_{i,N_P+N_D}(\infty))^T$ will be obtained by the walker. For convenience, we denote the node set DPD in WPDIN as the IS. Therefore, we can redefine the stable probability $R_i(\infty)$ as $R^{IS}{}_i(\infty)$. However, through observing $R^{IS}{}_i(\infty)$, it is easy to find that the walker will stop its random walking only after the walking probability vector calculated at each node in WPDIN is stable. In the face of

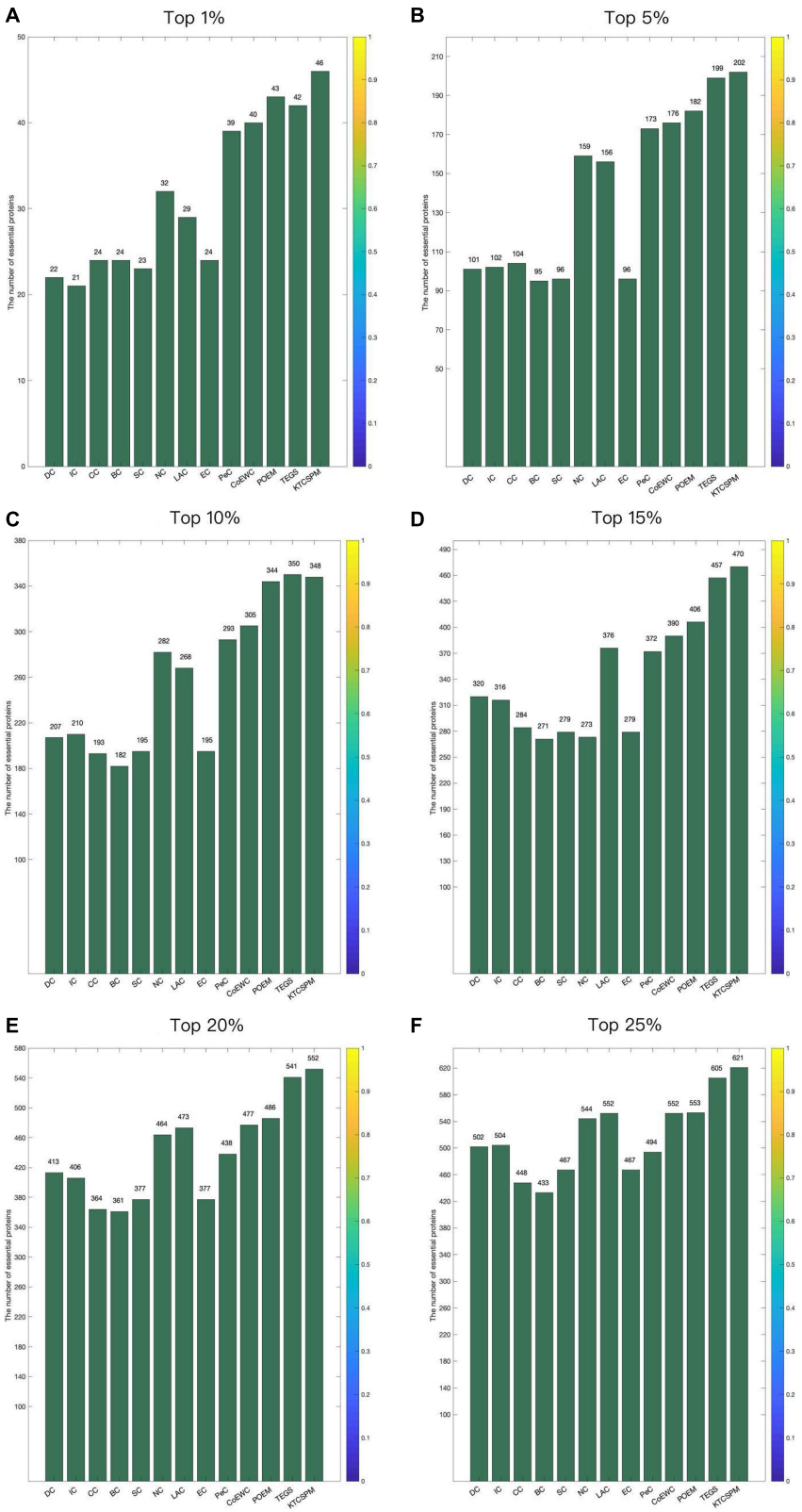

FIGURE 3 | (A) Top 1% ranked proteins. **(B)** Top 5% ranked proteins. **(C)** Top 10% ranked proteins. **(D)** Top 15% ranked proteins. **(E)** Top 20% ranked proteins. **(F)** Top 25% ranked proteins. In this Figure shows the predictive accuracy between KTCSPM and 12 competitive methods including IC, CC, BC, SC, NC, LAC, EC, PeC, CoEWC, POEM, and TEGS.

large data, this mechanism is obviously very time-consuming. Hence, in order to speed up the convergence speed of KTCSPM and reduce the experimental execution time, based on the concept of KTCS defined above, when constructing the vector $R_i(t) = (R_{i,1}(t), R_{i,2}(t), \ldots, R_{i,j}(t), \ldots, R_{i,N_P+N_D}(t))^T$ at the node pd_i, if the *j-th* node $pd_j \in KTCS\ (pd_i)$ in WPIND, then $R_{i,j}(t)$ will be remained unchanged, otherwise we will redefine $R_{i,j}(t) = 0$. Thus, the walking probability vector at pd_i will be changed to ${R_i}^{KTCS}(t)$ and the stable walking probability at pd_i will be changed to ${R_i}^{KTCS}(\infty)$. Obviously, the stable state of ${R_i}^{KTCS}(\infty)$ can be achieved faster than that of ${R_i}^{IS}(\infty)$. However, considering that there may be some nodes not belonging to *KTCS* (pd_i) but relating to the target, therefore, in order to avoid any omissions, at any given node pd_i in WPIND, we will construct a novel final stable walking probability vector $R_i^{ANS}(\infty) = (R_{i,1}^{ANS}(\infty), R_{i,2}^{ANS}(\infty), \ldots, R_{i,j}^{ANS}(\infty), \ldots, R_{i,N_P+N_D}^{ANS}(\infty))^T$ by combining ${R_i}^{IS}(\infty)$ with ${R_i}^{KTCS}(\infty)$ as follows:

$$R_i^{ANS}(\infty) = \frac{R_i^{IS}(\infty) + R_i^{KTCS}(\infty)}{2} \tag{19}$$

Step 5: For any protein node p_i in WPIND, according to the final stable walking probability vector $R_i^{ANS}(\infty)$ and the initial protein score *Initial_Score* (p_i) obtained above, it is obvious that a novel final feature score *Final_Score* (p_i) can be calculated as follows:

$$Final\_Score(p_i) = \frac{\sum_{j=1}^{N_P+N_D} R_{i,j}^{ANS}(\infty) + Initial\_Score(p_i)}{2} \tag{20}$$

Algorithm KTCSPM

Input

Original PPI network, original protein-domain network, domain data, subcellular data, orthologous data, and the proportion regulation parameters α.

Output

Proteins final score *Final_Score*(p_i).

Step 1: Establishing the heterogeneous network according to formulas (1–7);

Step 2: Calculating proteins initial score by orthologous data and subcellular data according to formulas (8–11);

Step 3: Establishing the KTCS according to formulas (12, 13);

Step 4: Establishing the transition probability matrix W according to formula (14);

Step 5: Calculating a stable walking probability vector $R_i(t)$ according to formulas (15–17);

Step 6: Establishing stable walking probability matrix K (∞) according to formula (18); and

Step 7: Outputting the final score of protein according to formula (19);

RESULTS

Experimental Data

In this section, extensive experiments will be done to compare KTCSPM with representative methods. And during experiments, the domain data is downloaded from the Pfam database (Bateman et al., 2014). The subcellular location data is derived from the COMPARTMENTS database (Binder et al., 2014), in

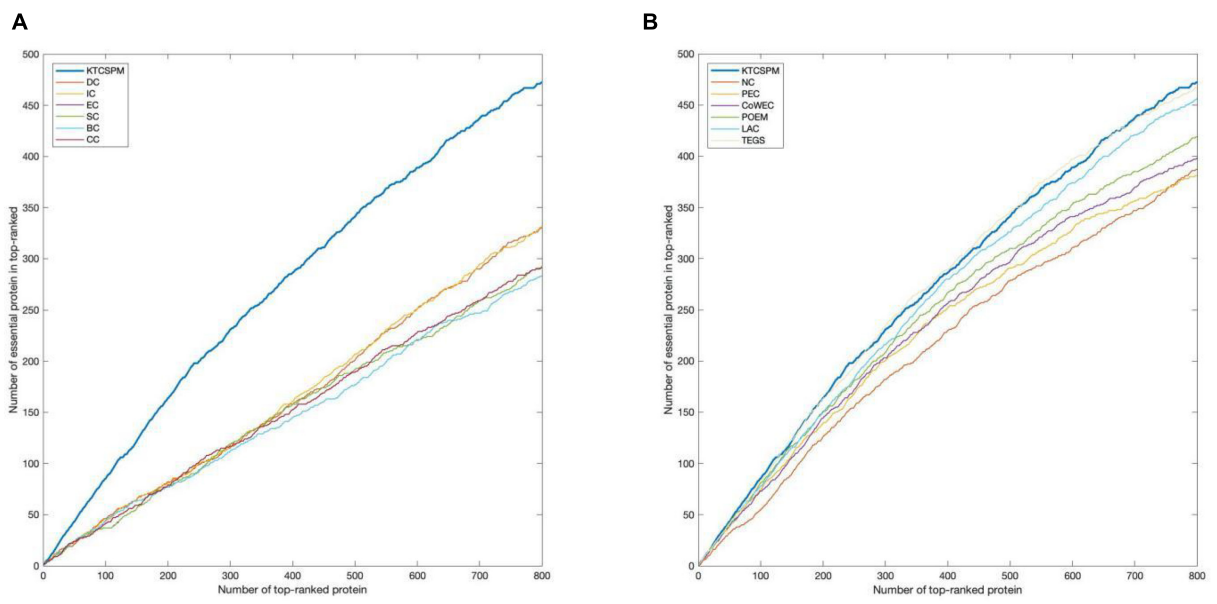

FIGURE 4 | The comparison results between key target convergence sets based prediction model (KTCSPM) and 12 competitive methods based on the DIP database under the jackknife methodology. **(A)** Comparison results between KTCSPM and DC, IC, EC, SC, BC, and CC. **(B)** Comparison results between KTCSPM and NC, PeC, CoWEC, POEM, LAC, and TEGS.

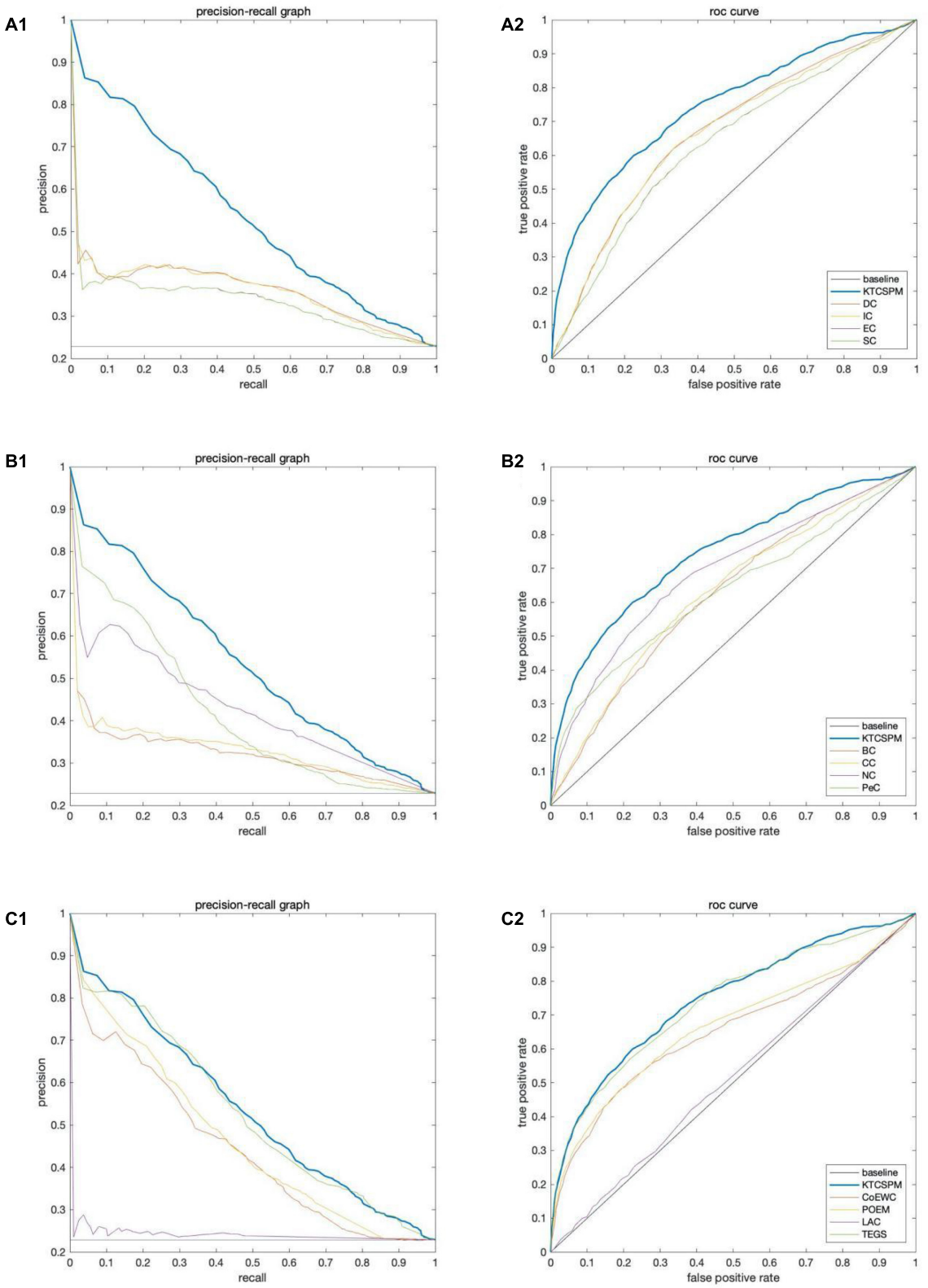

FIGURE 5 | Comparisons of performance between key target convergence sets based prediction model (KTCSPM) and 12 competitive models under the PR curve and ROC curve based on the DIP database. Panels **(A1–C1)** are comparison results of PR curves between KTCSPM and 12 competitive models. Panels **(A2–C2)** are comparison results of ROC curves between KTCSPM and 12 competitive models.

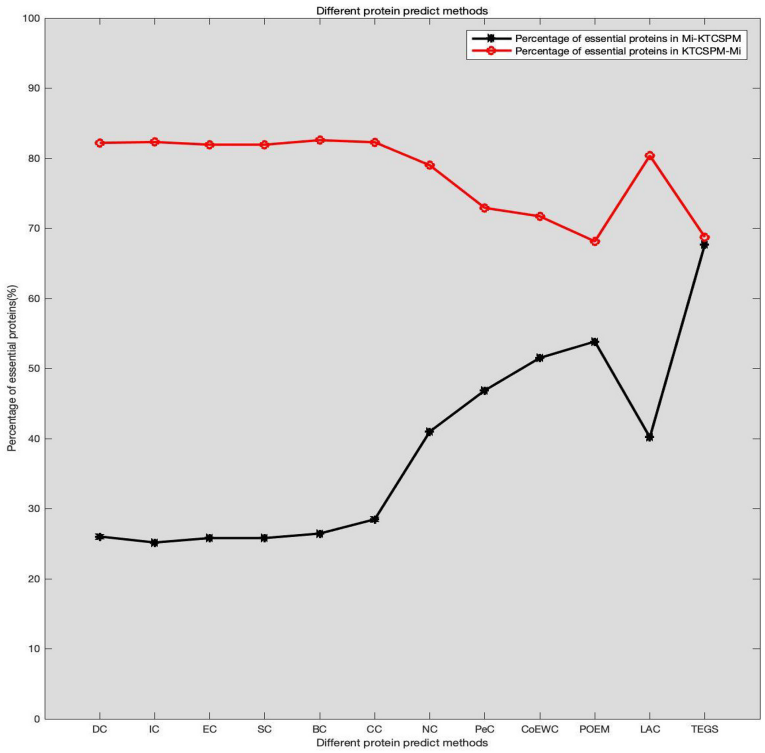

FIGURE 6 | Comparison results between KTCSPM and 12 competing methods, where the *X* axis denotes the competing methods including DC, IC, EC, SC, BC, CC, NC, PEC, COEWC, POEM, LAC, TEGS, and the *Y* axis represents the proportion of true essential proteins predicted by each method.

which, the following classifications of the subcellular interstitium related to the basic proteins of eukaryotic cells are included: Golgi bodies, endoplasm, cytoplasm, cytoskeleton, vacuoles, endosomes, mitochondria, plasma, peroxomes, and nuclei, etc. Besides, the reference bases of the essential genes of Scerevisiae are collected from MIPS (Mewes et al., 2006), SGD (Cherry et al., 1998), DEG (Zhang and Lin, 2009), and SGDP (Saccharomyces Genome Deletion Project, 2012). In the dataset downloaded from the DIP database, there are 5,093 proteins in total, in which, 1,167 are essential and 3,526 are non-essential. In the dataset downloaded from the GAVIN database, there are a total of 1,855 proteins, in which, 714 are essential proteins.

Comparison Between KTCSPM and Competitive Methods

In order to verify the predictive performance of KTCSPM, in this section, we will compare it with several representative methods such as DC (2001), IC (1989), And So-called Centrality (CC) (2014), Bee-tweenness Centrality (BC) (2005), SC (2003), NC (2005), PeC (2012),LAC (2011), CoEWC (2014), POEM (2017), and TEGS (2019) based on the DIP database and the Gavin database separately. **Figure 3** shows the comparison results between KTCSPM and these competitive methods. From observing **Figure 3**, it is obvious that the prediction accuracy of KTCSPM is significantly better than that of all these competing methods in from top 1 to 25% predicted essential proteins. In particular, KTCSPM can achieve a reliable prediction accuracy rate of 90.21% in the top 1% ranked key proteins.

Validation With Jackknife Methodology

For a comprehensive and accurate comparison, in this section, we will adopt the Jackknife methodology (Holman et al., 2009) to compare the predictive performances between KTCSPM and above mentioned competing methods. Experimental results are shown in **Figure 4**, from which, it can be clearly seen that the jackknife curve of KTCSPM is higher than that of all these state-of-the-art predictive methods. Although in **Figure 4B**, the jackknife curves of KTCSPM and TEGS have multiple intersections, however, when the number of ranked proteins is bigger than 600, the predictive results of KTCSPM will become continuously higher than that of TEGS. Therefore, according to both **Figures 4A,B**, we can draw a conclusion that KTCSPM can achieve better predictive performance than all these representative methods in predicting essential proteins.

Validation by Precision-Recall Curves and ROC Curves

In this section, ROC curve (receiver operating characteristic) and precision-recall curves (PR) will be adopted to measure the performance of KTCSPM. Researches show that the larger the area under the ROC curve (AUC), the better the model performance, and in addition, when AUC = 0.5, the model

TABLE 1 | Comparison results between KTCSPM and 11 state-of-the-art methods including DC, IC, CC, BC, NC, EC, PeC, CoEWC, ION, and POEM based on the Gavin database, where the Gavin database consists of 1,855 essential proteins.

Method	1%(19)	5%(93)	10%(196)	15%(279)	20%(371)	25%(464)
DC	7	36	101	158	222	264
IC	16	55	119	163	213	254
CC	11	45	93	135	180	221
BC	9	40	85	122	162	201
SC	9	36	87	130	190	240
NC	11	51	123	170	213	259
EC	0	38	94	134	166	209
PeC	15	69	142	193	238	285
CoEWC	16	69	136	190	237	275
ION	17	73	150	207	263	312
POEM	17	74	148	199	249	296
KTCSPM	17	75	160	216	269	315

TABLE 2 | Influence of the parameter α on prediction accuracy of KTCSPM based on the DIP database.

Rank α	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Top 1%	46	46	46	46	46	45	45	44	45
Top 5%	200	200	202	201	200	199	202	200	202
Top 10%	347	247	348	346	340	347	348	350	348
Top 15%	468	468	470	466	468	465	460	467	467
Top 20%	550	550	552	549	545	550	552	551	550
Top 25%	618	620	621	620	619	619	620	619	621

performance will be in a random state. Moreover, when PR curves are adopted to evaluate predictive models, more comprehensive feedbacks on performances of predictive models can be obtained by using different validation methods. And as a result, **Figure 5** shows the comparisons of performance between KTCSPM and 12 competitive prediction models under the PR curve and ROC curve separately. From the **Figures 5A1,2,B1,2**, it can be seen that when KTCSPM is compared with SC, EC, DC, IC, BC, CC, NC, PeC, the area under the PR curve (AUC), and ROC curve display results show that KTCSPM is superior. For these methods, by observing a3 and b3, it can be seen that when KTCSPM is compared with TEGS and POEM methods, the gap becomes smaller and there is overlap, but even so, the prediction performance of KTCSPM is still better than the 12 methods.

Analysis of the Differences Between KTCSPM and Competitive Methods

It can be seen from above descriptions that KTCSPM can achieve satisfactory predictive effects. In this section, we will further analyze the differences between KTCSPM and 12 competing methods by calculating the number of overlaps of first 200 predicted proteins. comparison results are shown in **Figure 6**, where Mi represents one of these 12 competitive methods, | KTCSPM-Mi| denotes the number of proteins detected by KTCSPM but not by Mi, | Mi-KTCSPM| means the number of proteins detected by Mi but not by KTCSPM. Obviously, according to the curve trends in **Figure 6**, we can see that the ratio of essential proteins predicted by KTCSPM is much higher than that predicted by anyone of these 12 competing methods, which means that KTCSPM can screen out more essential proteins not found by Mi, and demonstrates that KTCSPM can achieve much better predictive performance as well.

Prediction Performance of KTCSPM Based on the Gavin Database

In this section, in order to further verify the adaptability of KTCSPM, we will further compare it with 11 competitive methods based on the Gavin database, and comparison results are shown in the following **Table 1**.

Effects of Parameters on Performance of KTCSPM

In this section, we will estimate the effects of parameters on the prediction performance of KTCSPM. First, as for the parameter γ_p in Equation (1), we will set its value to one based on precedents (Twan et al., 2011). However, as for the parameter in Equation (17), as illustrated in **Table 2**, we will set its value from 0.1 to 0.9, and evaluate its impacts on the prediction performance of KTCSPM. Through observing **Table 2**, it is easy to see that when is set to 0.3, KTCSPM can achieve the best prediction effect. Moreover, it can be clearly seen that KTCSPM remains robust to different values of, which means that KTCSPM is not sensitive to the values of α.

DISCUSSION

It is time consuming and energy consuming to predict essential proteins through traditional biological experiments, so it has become a hot topic in the field of bioinformatics to build mathematical models to predict essential proteins. In this manuscript, a new prediction model called KTCSPM is proposed, in which, a weighted PDI network constructed by integrating a weighted PPI network and a weighted DDI network first, and then, based on the concepts of KCS and IS, a predictive method is further designed to infer potential key proteins in the weighted PDI network based on the random walk with restart. Finally, extensive experiments have demonstrated the predictive superiority of KTCSPM. At present, some methods have been proposed to infer potential disease related miRNAs such as RWRMDA (Chen et al., 2012), RLSMDA (Chen and Yan, 2014) and RBMMMDA (Chen et al., 2015), in the future, KTCSPM may also be applied to predict potential associations between miRNAs, and diseases.

CONCLUSION

In this manuscript, the main contributions are as follows: (1) A novel weighted PDI network is designed by combining a weighted PPI network with a weighted DDI network. (2) The concept of network distance is introduced, and the KTCS and the IS are established for nodes in the weighted PDI network. (3) Based on the concepts of KTCS and IS, an improved random

walk with restart algorithm is proposed to recognize essential proteins. By comparing with existing state-of-the-art predictive models, it is proved that KTCSPM can achieve better predictive performance in detecting essential proteins.

DATA AVAILABILITY STATEMENT

The datasets presented in this study can be found in online repositories. The names of the repository/repositories and accession number(s) can be found in the article/**Supplementary Material**.

AUTHOR CONTRIBUTIONS

JP conceived this research. JP, LW, and LK were improved on the basis of the original design model. JP and ZZ wrote the manuscript. YT and ZC provided advice and supervision on the research. All authors contributed to the article and approved the submitted version.

FUNDING

This research was supported by the National Natural Science Foundation of China (No. 61873221), the Research Foundation of Education Bureau of Hunan Province (No. 20B080), and the Natural Science Foundation of Hunan Province (Nos. 2018JJ4058 and 2019JJ70010).

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/fgene.2021.721486/full#supplementary-material

REFERENCES

Athira, K., and Gopakumar, G. (2020). An integrated method for identifying essential proteins from multiplex network model of protein-protein interactions. *J. Bioinform. Comput. Biol.* 18:2050020. doi: 10.1142/S0219720020500201

Bateman, A., Coin, L., Durbin, R., Finn, R. D., Hollich, V., Griffithsjones, S., et al. (2014). The pfam protein families database nucleic acids res. *Nucleic Acids Res.* 32, D138–D141. doi: 10.1093/nar/gkh121

Binder, J. X., Sune, P. F., Kalliopi, T., Christian, S., O'DonoghueSeán, I., Reinhard, S., et al. (2014). Compartments: unification and visualization of protein subcellular localization evidence. *Database J. Biol. Datab. Curat.* 2014:bau012. doi: 10.1093/database/bau012

Chen, X., and Yan, G. Y. (2014). Semi-supervised learning for potential human microRNA-disease associations inference. *Sci. Rep.* 4:5501. doi: 10.1038/srep05501

Chen, X., Huang, Y. A., You, Z. H., Yan, G. Y., and Wang, X. S. (2016). A novel approach based on KATZ measure to predict associations of human microbiota with non-infectious diseases. *Bioinformatics* 33, 733–739. doi: 10.1093/bioinformatics/btw715

Chen, X., Liu, M. X., and Yan, G. Y. (2012). RWRMDA: predicting novel human microRNA–disease associations. *Mol. Biosyst.* 8, 2792–2798. doi: 10.1039/c2mb25180a

Chen, X., Yan, C. C., Zhang, X., Li, Z., Deng, L., Zhang, Y., et al. (2015). RBMMMDA: predicting multiple types of disease-microRNA associations. *Sci. Rep.* 5:13877. doi: 10.1038/srep13877

Cherry, J. M., Adler, C., Ball, C., Chervitz, S. A., Dwight, S. S., and Hester, E. T. (1998). SGD: saccharomyces genome database. *Nucleic Acids Res.* 26, 73–79. doi: 10.1093/nar/26.1.73

Chua, H. N., Sung, W. K., and Wong, L. (2006). Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics* 24:452. doi: 10.1093/bioinformatics/btm609

Estrada, E. (2006). Protein bipartivity and essentiality in the yeast protein-protein interaction network. *J. Proteome Res.* 5, 2177–2184. doi: 10.1021/pr060106e

Estrada, E., and Rodriguez-Velazquez, J. A. (2005). Subgraph centrality in complex networks. *Phys. Rev. E Statist. Nonlin. Soft Mat. Phys.* 71:056103. doi: 10.1103/PhysRevE.71.056103

Fan, Y., Tang, X., Hu, X., Wu, W., and Ping, Q. (2017). Prediction of essential proteins based on subcellular localization and gene expression correlation. *Bmc Bioinform.* 18:470.

Gabriel, O., Thomas, S., Kristoffffer, F., Tina, K., David, N. M., Sanjit, R., et al. (2010). In Paranoid 7: new algorithms and tools for eukaryotic orthology analysis. *Nucleic Acids Res.* 38, D196–D203. doi: 10.1093/nar/gkp931

Gavin, A. C., Aloy, P., Grandi, P., Krause, R., Boesche, M., Marzioch, M., et al. (2006). Proteome survey reveals modularity of the yeast cell machinery. *Nature* 440, 631–636. doi: 10.1038/nature04532

Hahn, M. W., and Kern, A. D. (2004). Comparative genomics of centrality and essentiality in three eukaryotic protein-interaction networks. *Mol. Biol. Evol.* 22, 803–806. doi: 10.1093/molbev/msi072

Holman, A. G., Davis, P. J., Foster, J. M., Carlow, C. K., and Kumar, S. (2009). Computational prediction of essential genes in an unculturable endosymbiotic bacterium, Wolbachia of Brugia malayi. *BMC Microbiol.* 9:243. doi: 10.1186/1471-2180-9-243

Jeong, H., Mason, S., and Barabási, A. (2001). Lethality and centrality in protein networks. *Nature* 411, 41–42. doi: 10.1038/35075138

Joy, M. P., Brock, A., Ingber, D. E., and Huang, S. (2014). High-betweenness proteins in the yeast protein interaction network. *J. Biomed. Biotechnol.* 2005:96. doi: 10.1155/JBB.2005.96

Lei, X., Yang, X., and Wu, F. (2018). Artificial fish swarm optimization based method to identify essential proteins. *IEEE ACM Transact. Comput. Biol. Bioinform.* 18:1. doi: 10.1109/TCBB.2018.2865567

Li, J., Li, X., Feng, X., Wang, B., Zhao, B., and Wang, L. (2019). A novel target convergence set based random walk with restart for prediction of potential LncRNA-disease associations. *BMC Bioinform.* 20:3216. doi: 10.1186/s12859-019-3216-4

Li, M., Wang, J., Chen, X., Wang, H., and Pan, Y. (2011). A local average connectivity-based method for identifying essential proteins from the network level. *Comput. Biol. Chem.* 35, 143–150. doi: 10.1016/j.compbiolchem.2011.04.002

Li, M., Zhang, H., Wang, J. X., and Pan, Y. (2012). A new essential protein discovery method based on the integration of protein-protein interaction and gene expression data. *Bmc Syst. Biol.* 6:509. doi: 10.1186/1752-0509-6-15

Meng, Z., Kuang, L., Chen, Z., Zhang, Z., Tan, Y., Li, X., et al. (2021). Method for essential protein prediction based on a novel weighted protein-domain interaction network. *Front. Genet.* 12:645932.

Mewes, H. W., Frishman, D., Mayer, K. F. X., Munsterkotter, M., Noubibou, O., Pagel, P., et al. (2006). MIPS: analysis and annotation of proteins from whole genomes in 2005. *Nucleic Acids Res.* 34, D169–D172. doi: 10.1093/nar/gkj148

Min, L., Yu, L., Niu, Z., and Wu, F. X. (2017). United complex centrality for identification of essential proteins from PPI networks. *IEEE ACM Transact. Comput. Biol. Bioinform. (TCBB)* 14, 370–380. doi: 10.1109/TCBB.2015.2394487

Peng, W., Wang, J., Wang, W., Liu, Q., Wu, F. X., and Pan, Y. (2012). Iteration method for predicting essential proteins based on orthology and protein-protein interaction networks. *Bmc Syst. Biol.* 6:87. doi: 10.1186/1752-0509-6-87

Peng, W., Wang, J., Cheng, Y., Lu, Y., Wu, F., and Pan, Y. (2015). UDoNC: an algorithm for identifying essential proteins based on protein domains and protein-protein interaction networks. *Comput. Biol. Bioinform.* 12, 276–288. doi: 10.1109/TCBB.2014.2338317

Saccharomyces Genome Deletion Project (2012). Available online at: http://yeastdeletion.stanford.edu/ (accessed June 20, 2012).

Stephenson, K., and Zelen, M. (1989). Rethinking centrality: methods and examples. *Soc. Netw.* 11, 1–37. doi: 10.1016/0378-8733(89)90016-6

Twan, V. L., Nabuurs, S. B., and Elena, M. (2011). Gaussian interaction profile kernels for predicting drug–target interaction. *Bioinformatics* 21:3036. doi: 10.1093/bioinformatics/btr500

Vanunu, O., Magger, O., Ruppin, E., Shlomi, T., and Sharan, R. (2010). Associating genes and protein complexes with disease via network propagation. *PLoS Comput. Biol.* 6:e1000641. doi: 10.1371/journal.pcbi.1000641

Wang, J. X., Li, M., Wang, H., and Pan, Y. (2012). Identification of essential proteins based on edge clustering coefficient. *IEEE ACM Trans. Comput. Biol. Bioinform.* 9, 1070–1080. doi: 10.1109/TCBB.2011.147

Wuchty, S., and Stadler, P. F. (2003). Centers of complex networks. *J. Theoret. Biol.* 223, 45–53. doi: 10.1016/S0022-5193(03)00071-7

Xenarios, I., Salwinski, L., Duan, X. J., Higney, P., Sul-Min, K., and Eisenberg, D. (2002). DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Res.* 30, 303–305. doi: 10.1093/nar/30.1.303

Yu, H., Kim, P. M., Sprecher, E., Trifonov, V., and Gerstein, M. (2007). The importance of bottlenecks in protein networks: correlation with gene essentiality and expression dynamics. *PLoS Comput. Biol.* 3:e59. doi: 10.1371/journal.pcbi.0030059

Zhang, R., and Lin, Y. (2009). DEG 5.0, a database of essential genes in both prokaryotes and eukaryotes. *Nucleic Acids Res.* 37, D455–D458. doi: 10.1093/nar/gkn858

Zhang, W., Xu, J., and Zou, X. (2019). Predicting essential proteins by integrating network topology, subcellular localization information, gene expression profile and go annotation data. *IEEE ACM Transact. Comput. Biol. Bioinform.* 17, 2053–2061. doi: 10.1109/TCBB.2019.2916038

Zhang, W., Xu, J., Li, Y., and Zou, X. (2018). Detecting essential proteins based on network topology, gene expression data, and gene ontology information. *IEEE ACM Transact. Comput. Biol. Bioinform.* 15, 109–116. doi: 10.1109/tcbb.2016.2615931

Zhao, B., Wang, J., Li, M., Wu, F. X., and Pan, Y. (2014). Prediction of essential proteins based on overlapping essential modules. *IEEE Transact. Nano Biosci.* 13, 415–424. doi: 10.1109/TNB.2014.2337912

Zhao, B., Wang, J., Li, X., and Wu, F. X. (2016). Essential protein discovery based on a combination of modularity and conservatism. *Methods* 16, 54–63. doi: 10.1016/j.ymeth.2016.07.005

Zhao, B., Zhao, Y., Zhang, X., Zhang, Z., Zhang, F., Wang, L., et al. (2019). An iteration method for identifying yeast essential proteins from heterogeneous network. *BMC Bioinform.* 20:355.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.