**Abstract**— We propose a classifier system called iPFPi that predicts the functions of un-annotated proteins. iPFPi assigns an un-annotated protein $P$ the functions of GO annotation terms that are semantically similar to $P$. An un-annotated protein $P$ and a GO annotation term $T$ are represented by their characteristics. The characteristics of $P$ are GO terms found within the abstracts of biomedical literature associated with $P$. The characteristics of $T$ are GO terms found within the abstracts of biomedical literature associated with the proteins annotated with the function of $T$. Let $F$ and $F'$ be the important (dominant) sets of characteristic terms representing $T$ and $P$ respectively. iPFPi would annotate $P$ with the function of $T$, if $F$ and $F'$ are semantically similar. We constructed a novel semantic similarity measure that takes into consideration several factors, such as the dominance degree of each characteristic term $t$ in set $F$ based on its score, which is a value that reflects the dominance status of $t$ relative to other characteristic terms, using pairwise beats and loses procedure. Every time a protein $P$ is annotated with the function of $T$, iPFPi updates and optimizes the current scores of the characteristic terms for $T$ based on the weights of the characteristic terms for $P$. Set $F$ will be updated accordingly. Thus, the accuracy of predicting the function of $T$ as the function of subsequent proteins improves. This prediction accuracy keeps improving over time iteratively through the cumulative weights of the characteristic terms representing proteins that are successively annotated with the function of $T$. We evaluated the quality of iPFPi by comparing it experimentally with two recent protein function prediction systems. Results showed marked improvement.

**Index Terms**— Protein Function Prediction, Protein Annotation, Semantic Similarity, Biomedical Literature.

**I. INTRODUCTION**

Assigning biological functions to un-annotated proteins is one of the most important tasks in the post-genomic era due to the importance of proteins in various biological processes. Determining protein functions using biological experiments remain time consuming and expensive. Therefore, computational prediction systems have become an important alternative approach. However, this approach requires the development of accurate and reliable automatic predictors of protein function. Existing computational prediction approaches determine the functions of un-annotated proteins based on the functions of annotated proteins in the dataset. Most of these approaches predict protein functions from protein interaction networks [1, 2, 13, 17, 28], protein sequence [7, 12, 29], or protein structure [14, 25]. However, most of these approaches do not consider the interrelationships among functional terms and that a protein may be annotated with the functions of different terms.

Recent approaches take advantage of the exponential explosion of biomedical literatures. They predict protein function by extracting information from the literatures that describe the functions of other proteins. These approaches infer protein function using two ways, Information Extraction and Information Classification. Information extraction approaches (such as [11]) use features derived from literatures to represent proteins, even if the text features do not contain information about the functions of the proteins. Information extraction approaches [3, 4, 9, 10, 15] (which our approach in this paper is partially based on) extract phrases or terms from literatures that directly describe the function of the protein is involved. For example, AbXtract [4] extracts sentences that describe the protein function and then ranks them according to the statistical significance of the words in the sentences. Other approaches extract keywords from literature and associate these keywords with Gene Ontology (GO) annotation terms [10] through either dictionaries [15] or clustering techniques [9].

Inspired by our previous works [19-24], which studied the characteristics of GO annotation terms for the purpose of determining the relationships between genes and GO terms, we propose in this paper a novel approach for predicting protein function from the characteristics of GO annotation terms. Understanding the characteristics of GO annotation terms helps in understanding not only gene relationships, but also protein function. We implemented the proposed approach in a classifier system called iPFPi (Improving Protein Function Prediction Iteratively). In the framework of iPFPi, GO terms are used to annotate all protein functions. The classifier is trained by proteins known to be annotated with the functions of GO annotation terms. First, iPFPi measures the semantic similarities of the characteristics of an un-annotated protein $P$ and GO annotation terms. Then, it annotates $P$ with the functions of GO annotation terms that are semantically similar to it. This methodology avoids the problems of most current protein function prediction approaches, which do not consider the interrelationships among functional terms. We constructed a novel semantic similarity formula that measures the semantic similarities of the characteristics of an un-annotated protein and GO annotation terms. Every time an un-annotated protein $P$ is assigned the function of a GO annotation term $T$, the characteristics of $T$ are updated and optimized based on the characteristics of $P$. Thus, the accuracy of predicting the function of $T$ as the function of subsequent proteins keeps improving over time iteratively through the cumulative characteristics of proteins that are successively annotated with the function of $T$. 
The key differences between most current text-based protein function prediction approaches (such as Text-KNN [3]) and iPFPi are as follows: (1) most of these approaches predict the functions of an un-annotated protein \( P \) by measuring its similarity with other annotated proteins, while iPFPi predicts the function of \( P \) by measuring the semantic similarities of the most important characteristic terms for \( P \) and GO annotation terms, (2) iPFPi uses a novel semantic similarity measure that takes into consideration several factors such as the most important (dominant) characteristic terms representing an un-annotated protein and GO annotation terms, and (3) iPFPi updates the set of important characteristic terms representing a GO annotation term \( T \) based on the weights of the characteristic terms representing each protein that is recently annotated with the function of \( T \).

iPFPi can be used for identifying missing annotations as well as all annotations of a protein. Partially annotated proteins (i.e., proteins that have missing annotations) are very common. Some of the completely un-annotated proteins may not be associated with any PubMed abstract according to their entries in UniProtKB (i.e., textless proteins). To determine the functions of these proteins, iPFPi represents the characteristics of these proteins as text-based features using the following method described in [3] and EpiLoc [6]. If a textless protein has a homolog with associated text, iPFPi uses the text of the homolog to represent the protein. BLAST search is used for determining the list of homologs. The homologs are ranked by their e-values and the three ones with the lowest e-values are selected. Then, the combinations of the weights of the three feature vectors are calculated. We determine the degree of homology between the protein and its homologs, as follows. For each feature vector, we multiply its weights by the percentage of matched identical amino acids and then divide the results by three [3]. Finally, the protein is represented by the sum of the weighted feature vectors.

II. OUTLINE OF THE APPROACH

We adopt some of the Information Extraction techniques proposed in Text-KNN [3] for representing proteins by characteristic terms. Let \( F \) be the set of important characteristic terms representing an un-annotated protein \( P \). Let \( F' \) be the set of important characteristic terms representing a GO annotation term \( T \). iPFPi measures the semantic similarity of \( F \) and \( F' \), and annotates \( P \) with the function of \( T \) if \( F \) and \( F' \), are semantically similar. The following is an overview of our approach, in terms of the sequential processing steps taken by iPFPi to predict the functions of an un-annotated protein:

A. Selecting Proteins and GO Terms for Training iPFPi

We select a set of proteins annotated with reliable functions from a high-quality biological database such as UniProtKB/Swiss-Prot [5]. The selected set of proteins and the GO terms assigned to them will be used as a dataset for training iPFPi. Each of the selected proteins should satisfy the following two conditions: (1) should be annotated with at least one GO category, and (2) the entry of the protein in the biological database should have at least one reference to a PubMed abstract.

B. Representing a Protein by Characteristic Terms

We retrieve the abstracts of PubMed associated with the selected set of training proteins and referenced in the entry of the biological database. iPFPi extracts from these abstracts GO terms that are considered characteristic for each functional category. These characteristic terms will be used as text features to represent the training proteins. Our primary goal is to represent proteins using terms that are highly predictive of their potential functions [3]. We adopt the techniques proposed in GoPubMed [8] for locating GO terms in abstracts. Our GO term extraction algorithm relies on local string alignment to locate GO terms in abstracts. Since some authors refer to a GO term in abstracts without stating the term’s name/ID, iPFPi selects phrases in abstracts that contain at least a significant part of the semantic of the GO term. It employs a tokenizer and stemmer to align the sequence of words in abstracts and the names of GO terms. A term’s stemmed words are aligned against abstracts.

A GO term is considered a characteristic of a functional category \( f \), if its occurrence probability in abstracts associated with proteins whose functional category is \( f \) is statistically significantly different than its occurrence in abstracts associated with proteins belong to all other functional categories. First, we pre-process the abstracts to obtain a set of candidate characteristic terms. Then, we use the Z-Score statistical test to determine the characteristic terms by computing the differences of the terms’ occurrence probabilities across function classes [3, 6]. A GO term \( t \) is selected as a characteristic term for a function \( f \), if the value of the Z-score for \( t \) and \( f \) is greater than a specified value of standard deviation. The Z-score for \( t \) and \( f \) is defined as the distance between the raw score for \( t \) and the population mean in units of the standard deviation. The raw score for \( t \) is calculated by dividing the number of abstracts that contain \( t \) and associated with the proteins with the function \( f \) by the total number of abstracts associated with the proteins annotated with \( f \). The population mean is calculated by dividing the number of abstracts that contain \( t \) and associated with the proteins annotated with all functions \( f'' \) by the total number of abstracts associated with the proteins annotated with \( f'' \). A protein is represented by the union of all characteristic terms over all function classes.

C. Assigning Weights to Characteristic Terms

Each protein is represented by a vector of weights. Each weight represents the statistical significance of the occurrences of a characteristic term in the set of abstracts of PubMed associated with the protein. That is, each weight quantifies the likelihood of the association between a protein and a characteristic term based on the occurrences of the characteristic term in the set of abstracts of PubMed associated with the protein.

We model each protein \( P \) as a vector of weights \( V _ p = \{ (t_1, w_1^P), ..., (t_m, w_m^P) \} \), where: \( t_i \) denotes characteristic term \( i \) and \( w_i^P \) denotes a weight on \( t_i \). The weight \( w_i^P \) is the ratio of the number of occurrences of characteristic term \( i \) to the total number of occurrences of all characteristic terms in the set of abstracts associated with protein \( P \).
D. Representing a GO Annotation Term

Each GO annotation term $T$ is represented by a vector of weights. Each weight reflects the significance of a characteristic term in the set of abstracts associated with the proteins annotated with the function of $T$. That is, each weight for $T$ quantifies the likelihood of the association between a characteristic term and the set of proteins annotated with the function of $T$.

We model a GO annotation term $T$ as a vector of weights $V_T = \{(t_1, w'_1), \ldots, (t_m, w'_m)\}$, where: $t_i$ denotes characteristic term $i$ and $w'_i$ denotes a weight on $t_i$. The weight $w'_i$ is the ratio of the number of occurrences of characteristic term $i$ to the total number of occurrences of all characteristic terms in the set of abstracts associated with the proteins annotated with the function of $T$.

Example 1: Table 1 shows a dummy example illustrating how the weights of 10 characteristic terms $(t_1\ldots t_{10})$ for a GO annotation term $T$ are calculated based on the number of occurrences of these terms in 3 abstracts $(A_1\ldots A_3)$ associated with the proteins annotated with the function of $T$. Each entry in the table shows the number of occurrences of a characteristic term in an abstract. Each weight $w$ is the ratio of the number of occurrences of a characteristic term to the total number of occurrences of the 10 characteristic terms in the 3 abstracts.

<table>
<thead>
<tr>
<th>$t_1$</th>
<th>$t_2$</th>
<th>$t_3$</th>
<th>$t_4$</th>
<th>$t_5$</th>
<th>$t_6$</th>
<th>$t_7$</th>
<th>$t_8$</th>
<th>$t_9$</th>
<th>$t_{10}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A_1$</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$A_2$</td>
<td>0</td>
<td>1</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$A_3$</td>
<td>1</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>$w$</td>
<td>0.06</td>
<td>0.06</td>
<td>0.24</td>
<td>0.03</td>
<td>0.15</td>
<td>0.12</td>
<td>0.03</td>
<td>0.15</td>
<td>0.09</td>
</tr>
</tbody>
</table>

E. Assigning Functions to an Un-annotated Protein

Let $F$ be the important (dominant) set of characteristic terms representing a GO annotation term $T$, whose dominance scores are greater than a threshold. Let $F'$ be the important (dominant) set of characteristic terms representing an un-annotated protein $P$. We present in section III a semantic similarity formula that measures the semantic similarity of $F$ and $F'$.

iPFPi would annotate $P$ with the function of $T$, if the similarity of $F$ and $F'$ is greater than an empirically determined threshold $\delta_{sim}$.

F. Updating the Set of Dominant Characteristics Terms Representing a GO Annotation Term

We describe in section IV-B how iPFPi updates and optimizes the weights/scores of the characteristic terms representing a GO annotation term $T$ based on the weights of the characteristic terms representing an un-annotated protein $P$ that is recently annotated with the function of $T$. We also describe how the set of dominant characteristic terms for $T$ is updated accordingly. As a result, the accuracy of predicting the function of $T$ as the function of subsequent un-annotated proteins improves. This prediction accuracy keeps improving over time after each previously un-annotated protein is annotated with the function of $T$.

III. Constructing a Semantic Similarity Formula to Measure the Similarity of an Un-annotated Protein and a GO Annotation Term

iPFPi determines the semantic similarity of an un-annotated protein $P$ and a GO annotation term $T$ by matching the vectors of weights representing $P$ and $T$. We observe that we need to distinguish between the dominant (important) and non-dominant characteristic terms of $P$ and $T$ when measuring their semantic similarity. Usually, the dominant ones represent key characteristics and properties of $T$ and $P$, while the non-dominant ones may not. Considering the non-dominant characteristic terms in the similarity measure may result in a misleading similarity value, especially if the number of these terms is large, because their impact on the similarity value may be higher than those of the dominant ones. We construct through a series of refinements in subsections A-D below a semantic similarity measure that takes into consideration the above mentioned observation. Thus, the measure considers only dominant characteristic terms. It considers also some of the non-dominant characteristic terms for a GO annotation term $T$, if these terms are considered dominant for the un-annotated protein $P$, to which we need to find a function. However, each expression operand in the measure involving these non-dominant characteristic terms for $T$ is penalized to ensure that these terms have a lower impact on the similarity value.

A. Constructing an Initial Cosine-based Semantic Similarity Measure

Let $\text{sim}(P, T)$ be the semantic similarity of an un-annotated protein $P$ and a GO annotation term $T$. Equation 1 shows an initial version of the cosine-based semantic similarity formula employed by iPFPi for measuring $\text{sim}(P, T)$:

$$\text{sim}(P, T) = \frac{\sum_{t \in \langle V_P / V_T \rangle} (w_{P,t} - \bar{w}_P) (w_{T,t} - \bar{w}_T)}{\sqrt{\sum_{t \in \langle V_P / V_T \rangle} (w_{P,t} - \bar{w}_P)^2} \sqrt{\sum_{t \in \langle V_T / V_P \rangle} (w_{T,t} - \bar{w}_T)^2}}$$  (1)

- $w_{P,t}$: Weight of characteristic term $t$ in the vector of weights representing the un-annotated protein $P$.
- $w_{T,t}$: Weight of characteristic term $t$ in the vector of weights representing the GO annotation term $T$.
- $V_P$: Set of the characteristic terms representing protein $P$.
- $V_T$: Set of the characteristic terms representing GO term $T$.
- $V_T \cap V_P$: Set of the common characteristic terms representing both $P$ and $T$.
- $\bar{w}_P$: Mean weight of the common characteristic terms between $P$ and $T$ in the vector of $P$:

$$\bar{w}_P = \frac{\sum_{t \in \langle V_T \cap V_P \rangle} w_{P,t}}{|V_T \cap V_P|}$$
\* \( \tilde{w}_T \): Mean weight of the common characteristic terms between \( P \) and \( T \) in the vector of \( T \):

\[
\tilde{w}_T = \frac{\sum \tilde{w}_{r,t}}{|V_T \cap |V_P|}
\]

**B. Improving Equation 1 by Considering only the Dominant Characteristic Terms of GO Annotation Term \( T \)**

A characteristic term could be uninformative, if it has only few occurrences in abstracts and/or is assigned a high weight even though it is found in abstracts associated with many other classes. As described in section II-B, the \( Z \)-score values used for determining the characteristic terms representing a GO annotation term \( T \) are calculated based on the number of abstracts that contain these terms and associated with the proteins annotated with the function of \( T \). However, some of these abstracts may contain only one or a few occurrences of the characteristic terms. Moreover, some of these characteristic terms may be assigned a high weight even though they are found in abstracts associated with many other classes. Including uninformative terms could lead to misclassifying proteins of small function classes into the larger classes and vice versa. Therefore, equation 1 may give misleading results. To overcome this problem, we should refine the set of characteristic terms for \( T \) considered in the similarity equation by keeping only the dominant characteristic terms (i.e., the ones that have frequent occurrences in abstracts not associated with many classes) and excluding the uninformative ones.

Towards this, we assign a score to each characteristic term \( t \) for \( T \). The score reflects the dominance status of \( t \) relative to the other characteristic terms. First, we determine the pairwise beats and losses for each characteristic term contained in the abstracts associated with the proteins annotated with the function of \( T \). Characteristic term \( t_i \) beats characteristic term \( t_j \) if the number of times that the number of occurrences of \( t_i \) in the abstracts is greater than that of \( t_j \). Then, each characteristic term \( t_i \) assigned a score, which is the difference between the number of times \( t_i \) beats the other characteristic terms and the number of times it loses. Finally, the characteristic terms are ordered by their number of beats and losses (i.e., number of their pairwise beats minus number of their pairwise losses). The characteristic terms with the most beats are considered the dominant characteristic terms for \( T \). The remaining characteristic terms will be considered uninformative and will be excluded from the inclusion in the similarity equation.

**Definition 1 – A score of a characteristic term:** Let \( t_i \) denote the number of times the occurrences of characteristic term \( t_i \) in the set of abstracts associated with the proteins annotated with the function of a GO annotation term is greater than that of \( t_j \). Let \( S_{ih} \) denote the score of characteristic term \( t_i \) given the dominance relation \( > \) on the set of characteristic terms \( V_T \) for GO annotation term \( T \), the score \( S_{ih} \) equals:

\[
|t_i \in V_T : t_i > t_j| - |t_j \in V_T : t_i > t_j|
\]

The following are some of the distinctive features of the above scoring approach: (1) the sum of the scores of all characteristic terms is always zero, and (2) the highest and lowest possible scores for the characteristic terms are \((n-1)\) and \(-(n-1)\) respectively, where \( n \) is the number of characteristic terms. We also compute the normalized score \( \tilde{S}_i \) for each characteristic term \( t_i \). We compute \( \tilde{S}_i \) by adding the absolute of the most negative score to all scores and then normalizing the resulting values.

**Example 2:** Table 2 illustrates how the score \( S_i \) and normalized score \( \tilde{S}_i \) of the 10 characteristic terms presented in Example 1 are calculated based on their number of occurrences on the 3 abstracts shown in Table 1. The ranks of the 10 characteristic terms based on their weights in Table 1 are as follows: \( \{t_1, (t_5, t_6), t_6, t_6, t_1, t_2, t_1, t_2, t_1, t_2, t_1, t_2\} \). The ranks of the 10 characteristic terms based on their normalized scores in Table 2 are: \( \{t_1, t_0, t_2, t_1, t_3, t_0, t_0, t_0, t_0, t_0, t_0\} \). As can be seen, the ranks of some of the characteristic terms based on their weights are different than their ranks based on their normalized scores.

From the set of characteristic terms \( V_T \), the subset \( V_T^m \subseteq V_T \) is considered the dominant ones for GO annotation term \( T \), if every characteristic term \( t \in V_T^m \) satisfies the following:

1. dominates every characteristic term \( t' \in V_T \), \( t' \not\in V_T^m \) (the normalized score of \( t > \) the normalized scores of \( t' \))
2. Acquires a normalized score \( S_i \) greater than a threshold \( \beta \), which is the value lower than the mean normalized score by the standard error of the normalized mean

\[
\beta = \frac{1 - \frac{\sum_{t_j \in V_T} (S_i - \frac{1}{|V_T|})^2}{|V_T|}}{\sqrt{|V_T|}}
\]

**Table 2:** Beats/Loses Scores of the 10 Characteristic Terms Presented in Example 1, Calculated Based on Their Number of Occurrences in the 3 Abstracts Shown in Table 1. The Symbol “+” denotes that Characteristic Term \( t \) Beats Characteristic Term \( t' \) while “-” denotes that \( t \) Lost. “0” denotes that \( t \) and \( t' \) have the same number of beats and losses. \( S_i \) and \( \tilde{S}_i \) denote the Score and Normalized Score, respectively, of a Characteristic Term \( t_i \).

<table>
<thead>
<tr>
<th>( t_1 )</th>
<th>( t_2 )</th>
<th>( t_3 )</th>
<th>( t_4 )</th>
<th>( t_5 )</th>
<th>( t_6 )</th>
<th>( t_7 )</th>
<th>( t_8 )</th>
<th>( t_9 )</th>
<th>( t_{10} )</th>
<th>( t_{10} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>+</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>-</td>
<td>+</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-</td>
<td>-</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>+</td>
<td>+</td>
<td>+</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( S_i )</th>
<th>( \tilde{S}_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.12</td>
</tr>
<tr>
<td>0.2</td>
<td>0.1</td>
</tr>
<tr>
<td>0.08</td>
<td>0.17</td>
</tr>
<tr>
<td>0.15</td>
<td>0.08</td>
</tr>
</tbody>
</table>

**Definition 2 – Dominant characteristic terms:** Let \( V_T \) be the set of characteristic terms for a GO annotation term \( T \) and \( \tilde{S}_i \) the normalized score of a characteristic term \( t_i \). The subset \( V_T^m \subseteq V_T \) of the dominant characteristic terms for \( T \) with the maximal scores is given by: \( \{t_i \in V_T : \tilde{S}_i \geq \tilde{S}_{ij} \text{ for all } t_j \in V_T \text{ and } \tilde{S}_i > \beta \} \)
We adjusted equation 1 based on the concept of dominant characteristic terms described above as shown in Equation 3.

\[
sim(P, T) = \frac{\sum_{\forall t \in (T' \cap V_P)} \left( (w_{p,t} - \bar{w}_P) (w_{t,t} - \bar{w}_T) \right)}{\sqrt{\sum_{\forall t \in (T' \cap V_P)} \left( (w_{p,t} - \bar{w}_P) \right)^2} \sqrt{\sum_{\forall t \in (T' \cap V_P)} \left( (w_{t,t} - \bar{w}_T) \right)^2}}
\]  

(3)

- \( V''_T \): Set of the dominant characteristic terms for \( T \).
- \( \bar{w}''_P = \frac{\sum_{\forall t \in (T'' \cap V_P)} w_{p,t}}{|V''_T \cap V_P|} \); \( \bar{w}''_T = \frac{\sum_{\forall t \in (T'' \cap V_P)} w_{t,t}}{|V''_T \cap V_P|} \)

C. Improving Equation 3 by Considering the Dominant Characteristic Terms for \( T \) that are not Contained in the Abstracts Associated with the Un-annotated Protein \( P \)

Equation 3 overlooks the subset \( V''_T - V_P \) (i.e., the subset of dominant characteristic terms for \( T \) that is not contained in the abstracts associated with \( P \)). This may affect the accuracy of the similarity results, because the subset \( V''_T - V_P \) represents some of the key significant properties and characteristics for \( T \). The probability of the similarity result inaccuracy increases as the size of the subset \( V''_T - V_P \) increases. We observe that we can exclude the subset \( V''_T - V_P \) from the constraint of disregarding uncommon characteristic terms between \( T \) and \( P \). Towards this, we assign the subset \( V''_T - V_P \) a weight of zero in the vector of weights representing the un-annotated protein \( P \), since the subset does not represent characteristics and properties for \( P \). We adjusted Equation 3 accordingly as shown in Equation 4.

\[
sim(P, T) = \frac{\sum_{\forall t \in (T' \cap V_P)} \left( (w_{p,t} - \bar{w}_P) (w_{t,t} - \bar{w}_T) \right) + \sum_{\forall t \in U} \left( w_{p,t} \bar{w}_T \right) + \sum_{\forall t \in U} \left( w_{t,t} \bar{w}_P \right)}{\sqrt{\sum_{\forall t \in (T' \cap V_P)} \left( (w_{p,t} - \bar{w}_P) \right)^2} + \sum_{\forall t \in U} \left( (w_{p,t} \bar{w}_P) \right)^2 + \sum_{\forall t \in U} \left( (w_{t,t} \bar{w}_P) \right)^2}}
\]  

(4)

- \( U = V''_T - V_P \).
- \( \bar{w}''_P = \frac{\sum_{\forall t \in V''_T} w_{p,t}}{|V''_T|} \); \( \bar{w}''_T = \frac{\sum_{\forall t \in V''_T} w_{t,t}}{|V''_T|} \)
- \( W_{P,t} = 0 \) for \( \forall t \in (V''_T - V_P) \).

D. Improving Equation 4 by Considering the Dominant Characteristic Terms Representing the Un-annotated Protein \( P \)

We need to make sure that all the dominant characteristic terms representing the un-annotated protein \( P \) are considered in the similarity equation, because these terms represent the key characteristics and functional properties of \( P \). We consider each characteristic term \( t' \) for \( P \) a dominant characteristic term, if the following equation is satisfied:

\[
W_{P,t'} \geq \frac{1 - \sum_{\forall t \in V_P} (w_{P,t'} - 1/|V_P|)^2}{|V_P|}
\]

Some of these dominant characteristic terms for \( P \) may already be included in set \( V''_T \) in the similarity equation. We need to include in the equation also each dominant characteristic term \( t'' \) for \( P \) not included in set \( V''_T \). That is, we need to include each \( t'' \in V''_P - V''_T \), where \( V''_P \) is the set of dominant characteristic terms for \( P \). However, we need to penalize each expression operand in the equation involving \( t'' \) for \( T \) (i.e., each operand involving \( W_{T,t''} \)) to ensure that it will have a lower impact in the similarity value, since \( t'' \) is not a dominant characteristic term for \( T \). Moreover, we need to scale down these expression operands appropriately to account for the dominance rank of \( t'' \) for \( T \) among the list of characteristic terms for \( T \) to ensure that lower ranked characteristic terms indeed get higher penalty. Towards this, we penalize and scale down each expression operand involving \( t'' \) for \( T \) by a factor \( \text{decay}^{e-1} \), where \( \text{decay} \) is a parameter can be set to a value in the range 0 to 1. We set the exponent \( e \) to account for the rank of \( t'' \) for \( T \) among the list of characteristic terms for \( T \). We adjusted equation 4 accordingly as shown in equation 5.

\[
sim(P, T) = \frac{\sum_{\forall t \in (T' \cap V_P)} \left( (w_{p,t} - \bar{w}_P) (w_{t,t} - \bar{w}_T) \right) + \sum_{\forall t \in U} \left( w_{p,t} \bar{w}_T \right) + \sum_{\forall t \in U} \left( w_{t,t} \bar{w}_P \right)}{\sqrt{\sum_{\forall t \in (T' \cap V_P)} \left( (w_{p,t} - \bar{w}_P) \right)^2} + \sum_{\forall t \in U} \left( (w_{p,t} \bar{w}_P) \right)^2 + \sum_{\forall t \in U} \left( (w_{t,t} \bar{w}_P) \right)^2} + N \left( \sum_{\forall t \in (V''_P \cap V_T)} \left( (w_{T,t''} - \bar{w}_T) \right)^2 \times \text{decay}^{e-1} \right) + M \left( \sum_{\forall t \in (V''_P \cap V_T)} \left( (w_{T,t''} - \bar{w}_T) \right)^2 \times \text{decay}^{e-1} \right)
\]  

(5)

- \( N \) = \( \sum_{\forall t \in (V''_P \cap V_T)} \left( (w_{T,t''} - \bar{w}_T) \right)^2 \times \text{decay}^{e-1} \)
- \( M \) = \( \sum_{\forall t \in (V''_P \cap V_T)} \left( (w_{T,t''} - \bar{w}_T) \right)^2 \times \text{decay}^{e-1} \)
- \( V''_P \) : Set of dominant characteristic terms for \( P \).
- \( V''_T \) = \( V''_T - V''_P \)
- \( \bar{w}''_T = \frac{\sum_{\forall t \in (V''_P \cap V_T)} w_{t,t}}{|V''_P \cap V_T|} \); \( \bar{w}''_P = \frac{\sum_{\forall t \in (V''_P \cap V_T)} w_{p,t}}{|V''_P \cap V_T|} \)
- \( W_{T,t} = 0 \) for \( \forall t \in (V''_T - V_T) \).
- \( W_{P,t} = 0 \) for \( \forall t \in (V''_P - V_P) \).
IV. ASSIGNING FUNCTIONS TO AN UN-ANNOTATED PROTEIN AND UPDATING THE SCORES OF CHARACTERISTIC TERMS

A. Assigning Functions to an Un-annotated Protein

iPFPi predicts the functions of an un-annotated protein \( P \) based on the values of its semantic similarities with GO annotation terms, using equation 5. Most proteins are annotated with the functions of several GO annotation terms. Thus, \( P \) can belong to several function classes. Therefore, iPFPi orders GO annotation terms by their semantic similarities with \( P \), and it assigns to \( P \) the functions of the top ones, whose semantic similarities values are greater than an empirically determined threshold \( \delta_{sim} \). In our experiments, we determined the value of \( \delta_{sim} \) as 0.7.

iPFPi returns also a confidence score for each prediction made. The confidence score \( C(P) \) for a predicted function for an un-annotated protein \( P \) is calculated as follows. Let \( S(P) \) be the set of GO annotation terms predicted to annotate \( P \). Let \( S(P') \) be the set of GO annotation terms annotating each protein \( P' \) annotated with the function of a term \( \in S(P) \), where \( P' \neq P \). The confidence score \( C(P) \) is the average of the similarities between each GO annotation term \( T_P \in S(P) \) and each GO annotation term \( T_{P'} \in S(P') \), as shown in equation 6.

\[
C(P) = \frac{\sum_{T_P \in S(P)} \sum_{T_{P'} \in S(P'), \forall T_{P'} \neq T_P} \text{sim}(T_P, T_{P'})}{\sum_{T_P \in S(P)} \sum_{T_{P'} \in S(P')} |S(P')|}
\]

where \( \text{sim}(T_P, T_{P'}) \) is the semantic similarity of GO annotation terms \( T_P \) and \( T_{P'} \). We use the GO similarity measure proposed in [26] to compute \( \text{sim}(T_P, T_{P'}) \).

B. Updating the Scores of the Characteristic Terms for a GO Annotation Term

Every time an un-annotated protein \( P \) is annotated with the function of a GO annotation term \( T \), iPFPi updates and optimizes the current weights and scores of the characteristic terms for \( T \) based on the weights of the characteristic terms for \( P \). The set of dominant characteristic terms for \( T \) will also be updated accordingly. The weights of the characteristic terms of each un-annotated protein assigned the function of \( T \) would update and optimize the current weights and scores of the characteristic terms for \( T \) by updating: (1) the weights and number of beats/looses of the characteristic terms for \( T \) (recall Tables 1 and 2), and (2) the set of dominant characteristic terms for \( T \) (recall Definition 2). Thus, the accuracy of predicting the function of \( T \) as the function of subsequent un-annotated proteins improves. This prediction accuracy keeps improving over time after each previously un-annotated protein is annotated with the function of \( T \). That is, the prediction accuracy improves iteratively through the cumulative weights of the characteristic terms representing the proteins that are successively annotated with the function of \( T \).

The computation time complexity of updating and optimizing the current scores of characteristic terms for a GO annotation term is not expensive, especially when the updates are set to take place at certain update points (i.e., intervals of the number of function predictions). For example, in our experiments we set the intervals as 1000 for the biological process sub-ontology and 330 for the molecular function sub-ontology.

Example 3: Consider an un-annotated protein \( P \) represented by the vector of weights \( V_P = \{(t_1, 1), (t_2, 0), (t_3, 0), (t_4, 0), (t_5, 0), (t_6, 5), (t_7, 0), (t_8, 6), (t_9, 1), (t_{10}, 0)\} \). Consider that \( P \) is assigned the function of the GO annotation term \( T \) described in Example 1 and Table 1. After updating the vector of weights \( V_T \) and the set of normalized scores \( \overline{S_T} \) for \( T \) based on the weights of the characteristic terms for \( P \), \( V_T \) and \( \overline{S_T} \) will become as follows:

\[
V_T = \{(t_1, 0.07), (t_2, 0.04), (t_3, 0.17), (t_4, 0.02), (t_5, 0.11), (t_6, 0.2), (t_7, 0.02), (t_8, 0.24), (t_9, 0.09), (t_{10}, 0.04)\}
\]

\[
\overline{S_T} = \{(t_1, 0.13), (t_2, 0.07), (t_3, 0.16), (t_4, 0), (t_5, 0.03), (t_6, 0.2), (t_7, 0), (t_8, 0.2), (t_9, 0.16), (t_{10}, 0.06)\}
\]

As can be seen: (1) the ranks of most of the characteristic terms based on their normalized scores have changed after the update (recall Example 2 and Table 2), and (2) the ranks of some of the characteristic terms based on their weights may not correspond to their ranks based on their normalized scores (which is true before and after the update).

V. EXPERIMENTAL RESULTS

We experimentally evaluated the quality of iPFPi and compared it with Text-KNN [3] and CIA [27]. As described in sections 1 and 2, Text-KNN represents proteins by their characteristic terms (similar to iPFPi), but it predicts protein function using a k-nearest neighbor classifier. As for CIA, it predicts the function of a protein from protein-protein interaction (PPI) dataset using an iterative procedure. We implemented iPFPi in Java, run on Intel(R) Core(TM) i5 processor, with a CPU of 2.60 GHz and 4 GB of RAM, under Windows 7. A demo of iPFPi that identifies the Biological Process annotations of the complete Yeast protein dataset downloaded from [18] is available at:

http://ecesrvr.kustar.ac.ae:8080/iPFPi/

A. Compiling a Dataset for the Evaluation

1) Selecting Proteins for Testing and Training the Systems

We selected a fragment of GO graph from the biological process sub-ontology and a fragment of GO graph from the molecular function sub-ontology as sources for the evaluation dataset. The fragment from the biological process sub-ontology contains 70 GO terms annotating 583, 846 proteins. The fragment from the molecular function sub-ontology contains 30 GO terms annotating 603,438 proteins. From the proteins annotated with the functions of the 100 GO terms, we selected only the ones that: (1) are associated with at least one PubMed abstract according to their entries in UniProtKB [5], and (2) have experimental evidence code:
IC, IDA, EXP, IEP, TAS, IPI, IGI, IMP, or IC. The number of proteins that satisfy these two conditions is 78,908 (62,353 proteins annotated with the biological process sub-ontology and 16,555 proteins annotated with the molecular function sub-ontology). We downloaded the 100 GO terms and the 78,908 proteins annotated with their functions from [10]. We used them as a dataset for our experiments.

2) Selecting Characteristic Terms to Represent the Proteins used for Training iPFPi and Text-KNN
We retrieved 576,028 PubMed abstracts associated with the selected 78,908 proteins according to their entries in the UniProtKB/Swiss-Prot database [5]. We obtained from these abstracts the set of terms that are considered characteristics for each functional category. We selected the set of characteristic terms based on their Z-Score values as described in section II-B. We considered a term t as a characteristic term for a function f, if the value of the Z-score for t and f (using a 95% confidence level) is greater than “~1.96” standard deviation. We selected a total of 1,124 characteristic terms. We then computed the weight of each characteristic term as described in section II-C. The 1,124 characteristic terms over all function classes are used to represent the training and testing proteins. The union of 712 characteristic terms is used to represent each of the 62,353 training/testing proteins annotated with the biological process sub-ontology. The union of 412 characteristic terms is used to represent each of the 16,555 training/testing proteins annotated with the molecular function sub-ontology.

B. Evaluating the Prediction Performance using the Selected Dataset
In subsection 1, we measure the prediction performance of the three systems for individual GO annotation terms using 5-fold cross-validation. In subsection 2, we measure the prediction performance of the three systems using cumulative-validation dataset. In the two evaluations, the protein datasets were divided into training and testing proteins. For iPFPi, each GO annotation term T is represented by a vector of the weights of the training proteins annotated with the function of T, as described in section II-D. For Text-KNN [3], the vectors of the weights of the training proteins are used for determining the k-nearest neighbors of the testing proteins. For CIA [27], the training proteins are used to build a network of protein-protein interaction (PPI) dataset based on BioGrid site (http://thebiogrid.org/). We considered the testing proteins un-annotated and submitted them to the three systems to determine their functions. We measure the prediction performance for an un-annotated protein P using the standard metrics of Recall, Precision, and F-value as follows:

\[
\text{Recall} = \frac{c_p}{n_p}, \quad \text{Precision} = \frac{c_p}{m_p}, \quad F\text{-value} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}
\]

- \(c_p\): Number of correctly predicted functions for \(P\).
- \(n_p\): Number of actual functions of \(P\).
- \(m_p\): Number of predicted function for \(P\).

1) Evaluating the Prediction Performance using 5-fold Cross Validation Dataset
We performed 5-fold cross-validation using the 78,908 protein dataset. The protein dataset is partitioned (at random) into 5 disjoint subsets. Each subset contains the same distribution of class instances as in the original 78,908 protein dataset. The three systems are evaluated five times, where at each time a different subset of the data is used for testing while the remaining four subsets are used for training the systems. We measured the Recall, Precision, and F-value of the three systems using: (1) the entire dataset, and (2) only the proteins annotated with the function of each GO term. The results are shown as follows. Figs. 1 and 2 show the overall average Recall, Precision, and F-value of the three systems. Fig. 3 shows the accuracy of predicting the functions of each set of GO terms located at the same average depth (level) in the Biological Process sub-ontology. Fig. 4 shows the accuracy of predicting the functions of each set of GO terms located at the same average depth (level) in the Molecular Function sub-ontology. Tables 5 and 6 in the Supplemental Material show the average depth (level) of each GO term in GO Graph and the accuracy of predicting the function of this term.

We also evaluated the three systems using protein-centric metrics. We followed CAFA [16] procedure for plotting precision-recall curve according to a sliding threshold scheme. Only predictions with confidence scores higher than threshold values \(i (0 < i < 1)\) are selected for evaluation. We used thresholds distributed evenly in the range \([0,1]\) at step size 0.01. At each threshold, we calculated precision and recall for each protein and also the average precision and recall on all the protein dataset. At each threshold \(i\), the Recall \(rc_i(t)\) and Precision \(pr_i(t)\) for each protein \(i\) are calculated as follows:

\[
pr_i(t) = \frac{\sum I(f \in P_i(t) \land f \in T_i)}{\sum I(f \in P_i(t))}
\]

\[
rc_i(t) = \frac{\sum I(f \in P_i(t) \land f \in T_i)}{\sum I(f \in T_i)}
\]

where: (1) \(T_i\) is the subset of the 100 GO terms described in subsection V-A-1 that is experimentally determined for protein \(i\), (2) \(P_i(t)\) is the set of GO terms predicted by a system for protein \(i\) with score greater than or equal to \(t\), (3) \(f\) is a functional term in the ontology, and (4) \(I(\cdot)\) is the standard indicator function. The overall Recall and Precision for protein \(i\) at threshold \(t\) are calculated as follows:

\[
pr(t) = \frac{1}{m(t)} \sum_{i=1}^{m(t)} pr_i(t) \quad \text{and} \quad rc(t) = \frac{1}{n} \sum_{i=1}^{n} rc_i(t)
\]

where \(m(t)\) is the number of proteins that have at least one prediction above \(t\) and \(n\) is the number of all proteins in the dataset.

Fig. 5 shows the precision-recall curves of the three systems for the Biological Process annotations (the top figure) and the Molecular Function annotations (the bottom figure).
As Fig. 5 shows, iPFPi yielded higher precision than the other two systems, especially on recalls higher than about 0.4. At recall 0.5, for instance, on average 37% of iPFPi’s predicted Biological Process annotations and 45% of iPFPi’s predicted Molecular Function annotations are correct compared to 16% and 23%, respectively, of CIA’s correct annotation predictions (Text-KNN yielded 0% correct annotation predictions at recall 0.5). As Fig. 5 shows, iPFPi, CIA, and Text-KNN can reveal ~70%, ~60%, and ~45%, respectively, of the true annotations for proteins. More discussion of the results is presented in section V-F.

2) Evaluating the Prediction Performance using Cumulative-Validation Dataset

We partitioned the 78,908 protein dataset at random into training and testing disjoint subsets as shown in Table 3.

<table>
<thead>
<tr>
<th></th>
<th># Training Proteins</th>
<th># Testing Proteins</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biological Process classes</td>
<td>52,353</td>
<td>10,000</td>
<td>62,353</td>
</tr>
<tr>
<td>Molecular Function classes</td>
<td>13,255</td>
<td>3,300</td>
<td>16,555</td>
</tr>
</tbody>
</table>

We then performed 10 evaluation runs over a set of training proteins that accumulates in each run successively. The number of testing proteins in each run is 1,000 from the Biological Process dataset and 330 from the Molecular Function dataset. After each run, the current set of testing proteins will be added to the current set of training proteins, and the accumulating set will be used to train the systems in the next run. That is, the set of training proteins in each run is the cumulative of the training and testing proteins of the previous runs. Thus, the set of training proteins accumulates successively. The initial set of training proteins is 52,353 for the Biological Process classes and 13,255 for the Molecular Function classes (recall Table 3). Table 4 shows the number of training and testing proteins in each run.
After each run, iPFPi updates and optimizes the current scores of the characteristic terms representing the GO annotation terms based on the weights of the characteristic terms representing the current set of testing proteins (recall section IV-B). Fig. 6 shows the Recall, Precision, and F-value of the 3 systems using the dataset of 62,353 proteins annotated with the Biological Process sub-ontology. Fig. 7 shows the Recall, Precision, and F-value of the three systems using the dataset of 16,555 proteins annotated with the Molecular Function sub-ontology.
C. Evaluating the Prediction Performance using the Complete set of Yeast Dataset

We performed 5-fold cross-validation using the complete 6,086 Yeast protein dataset downloaded from [18]. Table 8 in the Supplemental Material shows the Characteristic Terms of a sample of 30 Yeast proteins from the 6,086 proteins extracted from the PubMed abstracts associated with these proteins. We followed the same procedure described in section V-B.1. Figs. 8 and 9 show the results. Table 7 in the Supplemental Material shows a sample of the 6,086 proteins and their Biological Process annotations identified by iPFPi. The last column in the Table shows the missing annotations identified by iPFPi. We discovered that 63% of the proteins have missing annotations based on their published annotations in GO website [10] and UniProtKB/Swiss-Prot database [5]. A demo of iPFPi that identifies the Biological Process annotations of the complete Yeast protein dataset is available at: http://ecesrvr.kustar.ac.ae:8080/iPFPi/

![Fig. 8](http://ecesrvr.kustar.ac.ae:8080/iPFPi/)  
**Fig. 8:** The performance of predicting the Biological Process annotations of the 6,086 Yeast protein dataset described in section V-C.

![Fig. 9](http://ecesrvr.kustar.ac.ae:8080/iPFPi/)  
**Fig. 9:** The performance of predicting the Molecular Function annotations of the 6,086 Yeast protein dataset described in section V-C.

D. Evaluating a Basic Version of iPFPi that Assigns a Protein the Functions of GO Terms that Frequently Occur in its Abstracts

In the framework of iPFPi, GO terms extracted from the abstracts associated with proteins are considered characteristics of functional classes. These characteristic terms are used as text features to represent proteins and are not used to annotate the proteins with their functions. In this test, we aim at studying whether an un-annotated protein can be annotated with the functions of the protein’s characteristic terms. If so, this protein can be annotated with the functions of the GO terms contained in the abstracts of biomedical literatures associated with the protein. Towards this, we constructed a basic version of iPFPi and Text-KNN [3] (since both use the same Information Extraction techniques for representing proteins by their characteristic terms). This basic version assigns a protein the functions of the GO terms that have frequent occurrences in the abstracts associated with this protein. It uses the Z-Score statistical test to identify highly occurred GO terms in the abstracts associated with the protein. It directly assigns the protein the functions of the top GO terms with the highest Z-Score values. We measured the Recall and Precision of this basic version using the same datasets and procedures described in sections V-B-1 and V-C. The experimental results revealed that the basic version achieved the following: 0.12 Recall and 0.15 Precision for the Biological Process annotations, and 0.17 Recall and 0.18 Precision for the Molecular Function annotations. As can be seen, both iPFPi and Text-KNN [3] significantly outperform this basic version. We concluded that GO terms extracted from the abstracts associated with a protein can be considered as characteristics of this protein and may not be used to annotate the protein with their functions.

E. Discussion of the Results

As can be seen from Figs. 1-11 that iPFPi outperformed both Text-KNN and CIA. Based on the experimental results, we analyze below the impact of the key features of iPFPi on its prediction performance:

- **Representing proteins by characteristic terms:** Both iPFPi and Text-KNN represent proteins by characteristic terms. However, and as can be seen from the results in Figs. 1-11 that iPFPi significantly outperformed Text-KNN. The poor prediction performance of Text-KNN is attributed to its inclusion of uninformative characteristic terms in the representation of proteins. A characteristic term can be uninformative if it has only few occurrences in abstracts and/or is assigned a high weight even though it is found in abstracts associated with many other classes. The inclusion of such uninformative terms leads to misclassifying proteins of small function classes into the larger classes and vice versa.

  iPFPi overcomes this problem by: (1) adopting the concept of dominant characteristic terms (which are informative terms) and excluding the non-dominant ones (which are uninformative); iPFPi predicts the functions of an un-annotated protein $P$ by measuring the semantic similarities of the dominant characteristic terms representing $P$ and GO annotation terms, and (2) representing all proteins, collectively, annotated with the function of a GO annotation term by a single vector of characteristic terms, which significantly reduces or eliminates uninformative characteristic terms.

- **Using cumulative iterative prediction procedure:** As can be seen from the results using the cumulative-validation dataset shown in Figs. 6 and 7 that the prediction performance of iPFPi improves constantly as the size of training proteins increases. This is because every time an un-annotated protein $P$ is annotated with the function of a GO annotation term $T$, iPFPi updates and optimizes the current scores of the characteristic terms of $T$ based on the weights of the characteristic terms of $P$. Thus, the accuracy of predicting the function of $T$ as the function of subsequent
un-annotated proteins improves. This prediction accuracy keeps improving over time after each previously un-annotated protein is annotated with the function of $T$. That is, the prediction accuracy improves iteratively through the cumulative weights of the characteristic terms representing the proteins that are successively annotated with the function of $T$. As for CIA and Text-KNN, the accumulation of training proteins has no noticeable impact on their prediction accuracy.

### Predicting protein functions from the characteristics of GO annotation terms:
We observed from the results of predicting the functions of individual GO terms that as the number of training proteins annotated with the function of a term $T$ gets larger, iPFPi tends to predict the function of $T$ more accurately. iPFPi may not predict the functions of very small classes accurately (classes with fewer than about 100 training proteins). On the other hand, CIA tends to predict more accurately the functions of GO terms annotating very small number of training proteins. This is attributed to the fact that CIA orders functions based on their influences and gives higher influences to functions with smaller number of proteins annotated with them (see equation 3 in [27]). This is disadvantageous to CIA, since the size of training proteins gets larger over time as un-annotated proteins are assigned functions. As for iPFPi, as the set of training proteins annotated with the function of a GO annotation term $T$ gets larger, the set of dominant characteristic terms representing $T$ becomes more optimized and more accurate. This is because the larger the number of training proteins gets, the more accurate becomes the scores assigned to characteristic terms based on their number of beats and losses (recall Table 2).

### Using a comprehensive semantic similarity measure:
The semantic similarity measures employed by CIA and Text-KNN do not include mechanisms for filtering the noisy data of PPI dataset and neighbor protein dataset respectively. The consistent performance of iPFPi over CIA and Text-KNN shown in Figs. 1-11 is attributed, in part, to the semantic similarity measure adopted by iPFPi, which filters noisy data by disregarding non-dominant (uninformative) characteristic terms.

### VI. Conclusion
We proposed a classifier system called iPFPi that predicts the functions of un-annotated proteins. iPFPi represents an un-annotated protein $P$ by characteristic terms found within the abstracts of biomedical literature associated with $P$. It represents a GO annotation term $T$ by characteristic terms found within the abstracts of biomedical literature associated with the proteins annotated with the function of $T$. iPFPi annotates $P$ with the function of $T$, if the dominant characteristic terms for both $P$ and $T$ are semantically similar. iPFPi would update and optimize the scores of the characteristic terms of $T$ based on the weights of the characteristic terms of $P$. Thus, the accuracy of predicting the function of $T$ as the function of subsequent un-annotated proteins improves. We experimentally compared iPFPi with two recent protein function prediction systems, and found that measuring the semantic similarity of the dominant characteristic terms of $P$ and $T$ using a cumulative iterative procedure can significantly improve function prediction accuracy.

### References
[18] SGD (Saccharomyces Genome Database), Available at: http://www.yeastgenome.org/download-data/curat
Paul D. Yoo received his PhD in Engineering and IT from the University of Sydney (USyd) in 2008. He was a Research Fellow in the Centre for Distributed and High Performance Computing, at the USyd from 2008 to 2009, and PhD Researcher (Quantitative Analysis) at the Capital Markets CRC, administered by the Australia Federal Dept. for Education, Science and Training, from 2004 to 2008. He is currently an Assistant Professor of Intelligent Systems and Big Data Analytics in ECE dept., Khalifa University, UAE. His recent research is centered on the application of various computer science and mathematical methods to the discovery of the syntactic and semantic patterns in nucleic acid and amino acid sequences. Paul holds over 40 prestigious journal and conference publications and is currently actively involved in editorial board, technical program committees, and review panels of the intelligent systems/machine-learning areas for international conference and journal publications such as IEEE and ACM.

Mohammed Alzaabi is a PhD Candidate in the Department of Electrical and Computer Engineering at Khalifa University of Science, Technology, and Research, UAE. He obtained his MS degree in information security from Khalifa University in 2010. He received his BS degree in computer engineering from the same university in 2009. Mr. Alzaabi primary research interests include digital forensics, network security, and graph theory. Mohammed received the UAE m-government service award, which was given to the first winning team in in the Smart Application Contest organized by the UAE Ministry of Higher Education and Scientific Research - Abu Dhabi, May 2014. He also received the 2013 Young Emirati Research prize, which was organized by the UAE Ministry of Higher Education and Scientific Research - Abu Dhabi, 2013. He is a member of the IEEE.