# Enhanced Sequence-Based Function Prediction Methods and Application to Functional Similarity Networks

**Meghana Chitale and Daisuke Kihara**

**Abstract** After reviewing the underlying framework required for computational function prediction in the previous chapter, we discuss two advanced sequence-based function prediction methods developed in our group, namely the Protein Function Prediction (PFP) method and the Extended Similarity Group (ESG) method. PFP extends the traditional homology search by incorporating functional associations between pairs of Gene Ontology terms based on the frequencies of co-occurrences in annotation of the same proteins in the database. PFP also considers very weakly similar sequences to the query, thereby increases its sensitivity and ability to predict low resolution functional terms. On the other hand, ESG recursively searches the sequence similarity space around the query to find consensus annotations in the neighborhood. The last part of the chapter discusses the network structure of gene functional space built by connecting proteins with functional similarity. Function annotation was enriched by predictions by PFP. Similarity to structures of protein-protein interaction networks and metabolic pathway networks is discussed.

## Introduction

In the previous chapter we have seen that there is a strong need to develop accurate function prediction techniques to deal with the explosive growth of newly sequenced genomes. The basic approach used for more than a decade is based on homology based annotation transfer. The assumption underneath this approach is that proteins that are evolutionarily related are also functionally related [1]. In this chapter we describe two advanced function prediction techniques, PFP [2, 3] and ESG [4], developed by our group, which extend the conventional homology search methods.

D. Kihara (✉)
Department of Biological Sciences; Department of Computer Science; Markey Center for Structural Biology, College of Science, Purdue University, West Lafayette, IN 47907, USA
e-mail: dkihara@purdue.edu

D. Kihara (ed.), *Protein Function Prediction for Omics Era*,
DOI 10.1007/978-94-007-0881-5_2, 

## Conventional Sequence-Based Function Prediction Methods

Conventionally, computational protein function prediction is largely based on transferring the functional knowledge from sequences similar to the one being searched. A typical procedure would be to first use sequence homology searches, such as BLAST [5] FASTA [6], or SSEARCH [7] to identify similar sequences from a sequence database. Functional annotations of these homologous sequences were transferred to the query sequences based on the E-value of the searches. SSEARCH [7] is the implementation of the rigorous Smith Waterman algorithm [8], and thus is the most accurate among the three methods [9, 10]. Due to computational complexity of this task, two faster algorithms, BLAST [5] and FASTA [6], that work on approximating the search without missing obvious homologs, are more popular in the research community. PSI-BLAST [11] is another method, which is more sensitive than the aforementioned three methods, which iterates searches by using a sequence profile computed from a multiple sequence alignment obtained from the search from the previous round. Following the homology search, it is common to identify functional domains and motifs in the query sequences by searching against domain databases, like BLOCKS [12], InterPro [13], Pfam [14], PRINTS [15], ProDom [16], PROSITE [17], SMART [18], SUPERFAMILY [19], TIGRFams [20], and PROSITE [17]. For more details, refer to recent review articles [21–23].

However, as discussed in Chapter 1, there are many cases that open reading frames in newly sequenced genomes do not find close homologs in the database, which will result in no annotation to the proteins. This situation has motivated the development of advanced techniques for function prediction. These methods are designed to use sequence search results in a more complex setting for obtaining larger annotation coverage yet maintaining or improving the accuracy. A class of methods extend homology search tools to extract function information in terms of Gene Ontology (GO) terms from retrieved sequences. These include Goblet [24], OntoBlast [25], GOFigure [26], Gotcha [27], GOPET [28], and ConFunc [29]. Using controlled vocabulary is essential for computationally retrieving and summarizing functional terms from a database search.

## Protein Function Prediction (PFP) Method

Our group has developed two function prediction methods, the Protein Function Prediction (PFP) method [2, 3] and the Extended Similarity Group (ESG) method [4], both of which predict GO terms from PSI-BLAST search results. There are some technical commonalities between the two methods, however, they are different in their design concepts.

PFP (http://kiharalab.org/pfp.php) is designed to extend the conventional PSI-BLAST search to consider very weakly related sequences. In a conventional use of (PSI-) BLAST searches, only significantly similar sequences to the query, which

have a similarity score (e.g. E-value) above a predefined threshold value (typical E-value threshold values are 0.001 or 0.01), are considered for extracting function information. However, there are frequently cases where weakly similar sequences have common function to the query, even if they appear below the threshold value in a search result [2]. Common functions between weakly similar sequences may be of "low resolution", which are less specific terms and are generally at shallower positions in the hierarchical structures of functional vocabularies. Such functions might not be useful for designing biochemical experiments but will be valuable information in large-scale functional analysis, e.g. analyses of microarray data or protein-protein interaction data, when functional information is not available otherwise.

The main advantage of PFP is that it can predict low resolution functions even in the absence of apparent sequence similarity with the query sequence. It extracts functional information (GO terms) from weakly similar proteins with weights derived from the E-value and combines them to form consensus about function of a query protein. PFP also uses an association mining tool called Function Association Matrix (FAM) that captures the relations between pairs of GO terms in term of conditional probabilities of observing one annotation provided that the protein has another annotation.

## *PFP Algorithm*

PFP takes a query sequence as an input and predicts GO terms that are likely to annotate the sequence with a confidence score. It predicts GO terms in all the three categories, Molecular Function (MF), Biological Process (BP), and Cellular Component (CC). PFP first uses PSI-BLAST [11] to obtain similar sequences from a database with the E-value cutoff of 100. For each of the retrieved sequences, GO annotations are obtained from the PFPDB database, which combines GO annotations from Gene Onotology Association (GOA) [30] database, HAMAP [31], InterPro [13], Pfam [14], PRINTS [15], ProDom [16], PROSITE [17], SMART [18], and TIGRFams [20]. GO terms taken from each sequence are weighted and summed as follows:

$$s(f_a) = \sum_{i=1}^{N} \sum_{j=1}^{Nfunc(i)} \big((-\log(E\_value(i)) + b)P(f_a|f_j)\big), \tag{1}$$

where $s(f_a)$ is the final score assigned to the GO term $f_a$, $N$ is the number of similar sequences retrieved by PSI-BLAST, $Nfunc(i)$ is the number of GO terms annotating sequence $i$, $E\_value(i)$ is the E-value given to the sequence $i$, $f_j$ is a GO term annotating sequence $i$, and $b$ is the constant value, $2 = (\log_{10}100)$, which keeps the score positive. $P(f_a|f_j)$ is the association score for $f_a$ given $f_j$ obtained from the function association matrix (FAM).

FAM captures the co-occurrence of pairs of GO terms annotating the same protein in UniProt [32] database as the form of the conditional probability. FAM captures knowledge that is obvious to biologists but not reflected to annotations in the database. For example, a GO term in the MF category, *DNA binding* is frequently related to another GO term in the BP category, *regulation of transcription*. Thus, if we obtain a sequence hit with annotation *DNA binding* then the term *regulation of transcription* will obtain a share of the score from the association. Importantly, the relationship of these two GO terms cannot be captured by considering the GO hierarchy, because the two terms are on different trees.

FAM conditional probability score is obtained as follows:

$$P\left(f_a|f_j\right) = \frac{c(f_a,f_j) + \varepsilon}{c(f_j) + \mu \cdot \varepsilon'} \tag{2}$$

where $c(f_a,f_j)$ is number of times $f_a$ and $f_j$ are assigned simultaneously to each sequence in UniProt, and $c(f_j)$ is the total number of times $f_j$ appeared in UniProt, $\mu$ is the size of one dimension of FAM (i.e. the total number of unique GO terms), and $\epsilon$ is the pseudo-count.

Thus, the association strategy allows PFP to explore the functional space further from annotations obtained directly from sequence hits using PSI-BLAST, which helps developing consensus about low resolution function in the absence of strong hits.

Additionally, PFP makes use of the hierarchy of the GO terms (directed acyclic graph, DAG) by propagating scores to each parent term based on the number of gene products associated with parent term as compared to the child term, as shown in Eq. (3). Due to this scheme some low resolution functions can get high scores by summing the scores propagated from multiple child nodes, and thus helping PFP predict some annotations where no strong sequence similarity exists.

$$s(f_p) = \sum_{i=1}^{N_c} \left( s(f_{ci}) \left( \frac{c(f_{ci})}{c(f_p)} \right) \right), \tag{3}$$

where $s(f_p)$ is the score of the parent term $f_p$, $N_c$ is the number of child GO terms which belong to the parent term $f_p$, $s(f_{ci})$ is the score of a child term $ci$, and $c(f_{ci})$ and $c(f_p)$ is the number of known genes which are annotated with function term $f_{ci}$ and $f_p$ in the annotation database.

Finally, we compute the p-value significance scores for each prediction using the raw score distribution of each GO term obtained from a benchmarking dataset. Each of the p-values is associated with an expected accuracy score calculated at three different levels (correct predictions within 0, 2 and 4 edge distance on GO DAG) using the benchmarking dataset [2]. Since raw scores from Eq. (1) tend to be large for less specific terms, p-values and expected accuracy should be considered when selecting predictions done by PFP.

## *PFP Performance and Benchmarking*

PFP has been benchmarked in two papers. In the first paper [3], we have studied the prediction performance of PFP at different cutoff levels of E-value of PSI-BLAST search. Namely, sequence hits above each cutoff value were ignored mimicking situations that there are no significant hits up to the E-value. A dataset of 2,000 randomly selected proteins from UniProt was used for the benchmark. The prediction accuracy was measured in terms of the sequence coverage, which is the percentage of sequences in the benchmark set which are annotated with correct predictions. At all E-value cutoffs, PFP showed a significantly higher coverage over a simple transfer of annotations from the top scoring sequence retrieved by PSI-BLAST (top PSI-BLAST). At the E-value cutoff of 10 (i.e. only sequences with an E-value of 10 or larger are used), PFP showed nearly five times more coverage (50%) as compared to the top PSI-BLAST method. It was also shown that the FAM improved the sequence coverage by 5–20%. Interestingly, PFP showed a better coverage over the top PSI-BLAST even when no sequence hits were ignored. This indicates that taking consensus functions among sequence hits yields better prediction in general, since often sequences of significant similarity have different functions. These results indicate that PFP can very well utilize weakly similar proteins which do not share apparent sequence similarity with a query protein.

In another study [2] using a benchmark dataset of 120,260 proteins from 11 genomes, performance of PFP has been compared against two protein function prediction methods, GOtcha [27] and InterProScan [33], as well as the top PSI-BLAST in terms of the three GO category version of the funSim score [34] (Eq. (12) in Chapter 1). It was observed in the head to head comparison among PFP, Gotcha [27], and the top PSI-BLAST [11] that PFP significantly outperformed both methods at all E-value cutoffs used, winning around 60% of cases. We have also tested different parameter values thoroughly in the paper. In addition, the p-value of the raw PFP score and the relationship between the p-value and the accuracy was examined.

As discussed above, one of the main advantages of PFP lies in its ability to increase the annotation coverage as compared to conventional homology searches. Annotations to fifteen genomes showed that more than two third of unknown proteins in each genome were assigned molecular function term at a high confidence with an expected accuracy level of 80%.

The effect of PFP's annotation to less annotated genomes can be quite dramatic. As an illustration, functional enrichment by PFP for the protein-protein interaction (PPI) network of *Plasmodium falciparum* (malaria) is shown in Fig. 1. In the original annotations in the database 664 interactions have both interacting proteins annotated (fully annotated), one of the proteins is annotated in 1,358 interactions, while 824 have neither of interacting nodes annotated. Using PFP predictions with the expected accuracy of over 90%, the number of fully annotated interactions increased to 2,674. And the number of interactions where both interacting partners are unknown was dramatically reduced to 4. These annotations will be useful for biological understanding of protein interactions in the PPI network.

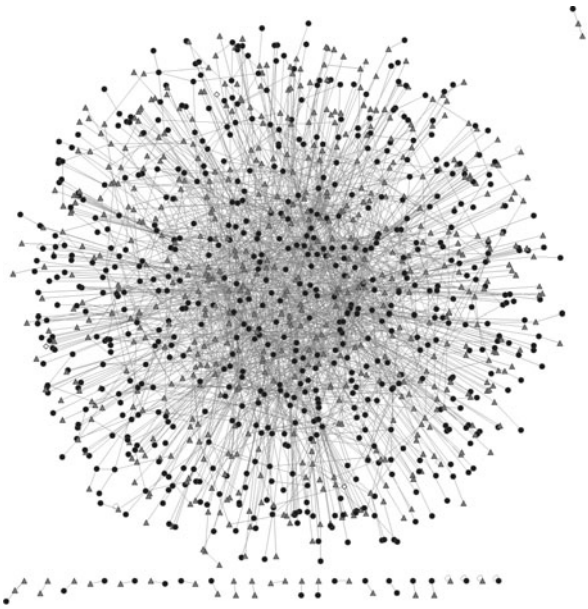

**Fig. 1** *P. falciparum* PPI network with following color coding for nodes – *black circles*: proteins annotated by PFP at high confidence (>80% confidence) in at least one GO category, *gray triangles*: proteins previously annotated in the database in at least one GO category, *white diamonds*: un-annotated proteins

## Extended Similarity Group (ESG) Method

The Extended Similarity Group (ESG) method [4] (http://kiharalab.org/esg.php) iterates PSI-BLAST searches by using sequences retrieved in a previous round as queries for the next round of search. GO terms taken from a retrieved sequence are weighted in a similar way as PFP, considering the E-value of the sequence. Since there are multiple rounds of searches, each round is weighted by another parameter.

### *The ESG Algorithm*

ESG begins with an initial PSI-BLAST [11] search from the query sequence $Q$, which will retrieve $N$ sequence hits, $S_1, S_2, \ldots S_N$ each with E-value $E_1, E_2, \ldots E_N$, respectively. The sequences are weighted by $W_i$, which considers the significance of E-value of sequence $S_i$ relative to the other sequences:

$$W_i = \frac{-\log(E_i) + b}{\sum_{j=1}^{N} \left\{ -\log(E_j) + b \right\}}, \tag{4}$$

where score, –log($E_i$), is shifted by a constant value $b$, which makes the score a non-negative value. Using the Eq. (4) assures that the weights to the N sequences sum up to 1. Using the weights $W_i$ assigned to each sequence, the probability of the

GO term $f_a$ annotating the query sequence $Q$ is defined as the sum of weights of $f_a$ that come from sequences annotated with $f_a$:

$$P_Q^d(f_a) = \sum_{i=1}^{N} W_i \cdot I_{Si}(f_a) \tag{5}$$

The function $I$ indicates whether the given sequence $S_i$ has annotation $f_a$:

$$I_{S_i}(f_a) = \begin{cases} 1 \text{ if } S_i \text{ has } f_a \text{ annotation} \\ 0 \qquad \text{otherwise} \end{cases} \tag{6}$$

The index $d$ on the left side of Eq. (5) denotes that function information comes from direct annotations to sequences. Later we formulate integration of the FAM, which captures associated GO terms rather than directly assigned GO terms to each sequence, in the ESG framework.

Now we extend this concept to multiple levels of PSI-BLAST searches by sharing the weights between levels using a weight parameter $v$. In the second round, each of the sequences $S_1, S_2, \ldots S_N$ retrieved in the first round are in turn used as a query. Suppose sequence $S_i$ obtains $N$ sequences by a PSI-BLAST run, each referred as $S_{ij}$. The weights for $S_{ij}$, $W_{ij}$ can be computed in a similar manner to Eq. (4). Combining the two level of searches,

$$P_Q^d(f_a) = \sum_{i=1}^{N} W_i \cdot P_{Si}^d(f_a) \tag{7}$$

$$P_{S_i}^d(f_a) = v \cdot I_{S_i}(f_a) + (1 - v) \cdot \sum_{j=1}^{N_i} W_{ij} \cdot I_{S_{ij}}(f_a) \tag{8}$$

Equation (7) is essentially the same as Eq. (5), representing that the score of a GO term $f_a$ for the query Q is contributed by sequences retrieved at the first level ($S_1$ to $S_N$). The weight $W_i$ is defined by Eq. (4). Equation (8) defines the score for $f_a$ for sequence $S_i$ as a combination of $I_{S_i}(f_a)$, which is sequence $S_i$'s annotation, and the second level search. The first and the second terms are weighted by a factor $v$. The equations can be recursively extended to multiple levels of searches to explore broader space around the query sequence.

The algorithm for the two level of the search is illustrated in Fig. 2. It shows the probability computations as described by Eqs. (7) and (8).

The FAM, which considers association of GO term pairs (Eq. (2)), can be integrated to the ESG algorithm. Equation (7) is replaced with the following equation, which states that now FAM is used for function annotation:

$$P_Q^{\text{FAM}}(f_a) = \sum_{i=1}^{N} W_i \cdot P_{S_i}^{\text{FAM}}(f_a) \tag{9}$$

For ESG with the second level search, Eq. (8) is modified to

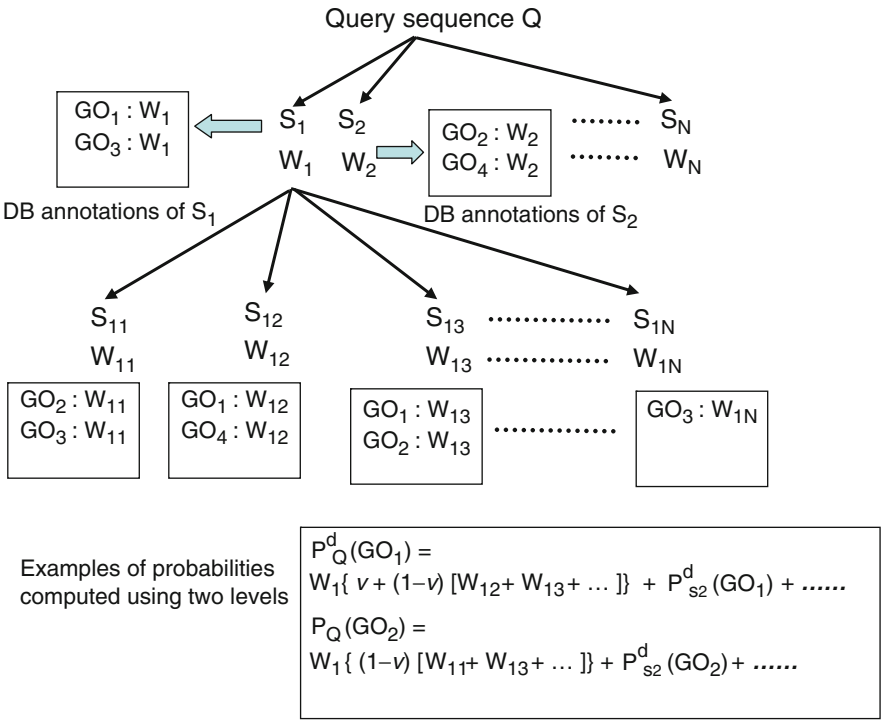


**Fig. 2** Probability computation in ESG for two levels

$$P_{S_i}^{\mathrm{FAM}}(f_a) = v\left\{I_{S_i}(f_a) + (1 - I_{S_i}(f_a)) \cdot \max\left(\sum_{j=1}^{N_{S_i}} P(f_a|f_j), 1\right)\right\} + (1 - v)\left\{\sum_{j=1}^{N_{ij}} W_{ij} \cdot P_{S_{ij}}^{\mathrm{FAM}}(f_a)\right\}, \tag{10}$$

where $N_{Si}$ is the number of GO terms annotating sequence $S_i$. The first and the second level searches are weighted by a factor $v$. The first term shows that in case $S_i$ is not directly annotated with $f_a$, the FAM is used to consider association of each GO term annotating $S_i$ to function $f_a$. The max operation is used to not to let the FAM-based score exceed 1. $P_{Sij}^{\mathrm{FAM}}(f_a)$ in the second term is expanded in the same way as the first term:

$$P_{S_{ij}}^{\mathrm{FAM}}(f_a) = I_{s_{ij}}(f_a) + (1 - I_{S_{ij}}(f_a)) \cdot \max\left(\sum_{j=1}^{N_{ij}} P(f_a|f_j), 1\right) \tag{11}$$

The formulation of the score (Eqs. (4), (5), (6), (7), (8), (9), (10), and (11)) provides a value ranging from 0 to 1.

## *Performance of ESG*

Performance of ESG has been benchmarked on a set of 2,400 protein sequences, which consists of 200 randomly selected proteins from twelve different genomes. The results of using two score cutoff values, 0.35, and 0.15, which were shown to provide a good balance of precision and recall, are shown in Fig. 3. Predicted GO terms were evaluated in terms of the funSim score with the three GO categories (Eq. (12) in chapter "Computational Protein Function Prediction: Framework and Challenges"). The FAM was not used and the search was iterated for two levels for these results. It was observed for all but one genome that the funSim scores of ESG are better than PFP. The average score of ESG was around 0.7 while that of PFP was around 0.6. Both of the methods showed superior performance as compared to the top PSI-BLAST method, which showed the average funSim score around 0.2.

Further, it was observed that ESG shows far better performance than PFP in terms of precision. ESG predicts a smaller number of GO terms as compared with PFP (average 7 GO terms are predicted by ESG while 60 terms by PFP), which generally reduces false positives, and results in an increased precision. The average precision for ESG was observed approximately 0.7 while that for PFP and top PSI-BLAST was around 0.10 on this benchmark dataset. Moreover, ESG showed a slightly better recall value than PFP, with 0.6 for ESG and 0.5 for PFP, respectively.

ESG has also been extended to incorporate the FAM, which has been shown to improve prediction recall with slightly reduced precision. For 200 *E. coli* proteins in the dataset, the recall increased from 0.773 to 0.810 by incorporating the FAM but

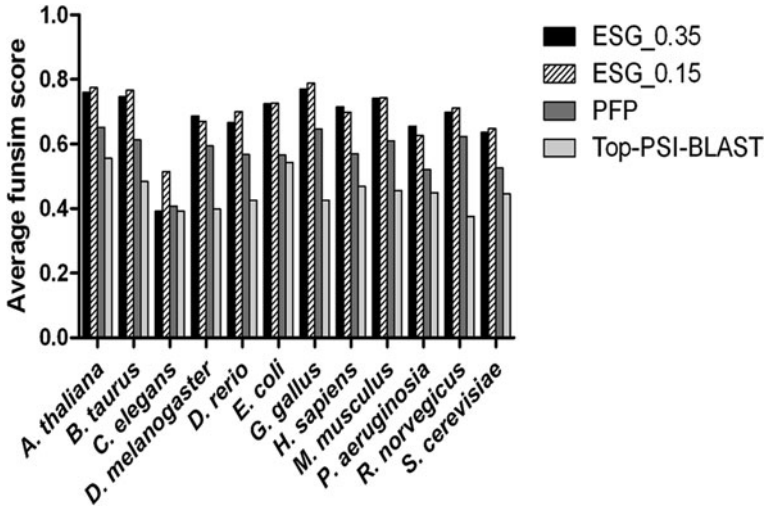


**Fig. 3** The average semantic similarity score on the benchmark dataset. Two probability cutoff values are used for ESG, 0.35 and 0.15. For the Top PSI-BLAST, GO terms are extracted from sequence hits with E-value of 0.01 or smaller (better). For the PFP, GO terms with no less than 80% expected accuracy were considered. (This figure is modified from fig. 3 in [4])

the precision decreased from 0.794 to 0.566. This effect is due to the increase of the average number of predicted terms by using FAM. Overall the results indicate that PFP and ESG have considerably improved the prediction accuracy for automated function prediction using sequence similarity search.

## *Difference between PFP and ESG*

Figure 4 illustrates difference of PFP and ESG with a conventional PSI-BLAST search. In the conventional PSI-BLAST search, only significantly similar sequences to the query (shown as the filled circle), e.g. within the E-value of 0.001 or 0.01 (dashed circle), are considered. In contrast, PFP extends the search to the E-value of 100 in the sequence similarity space, which results in more sensitive prediction. On the other hands, ESG iterates searches around the query and takes GO terms that consistently appear among the searches. Thus, ESG is designed to increase the precision of prediction. Both PFP and ESG outperform the conventional PSI-BLAST search in general, because annotations in some of closely similar sequences, which do not apply for the query, can be discarded by considering consensus among a larger number of sequences.

There is also a significant difference in the design of the score of PFP and ESG. In PFP, the raw score for each GO term is simply the sum of the scores computed from each sequences retrieved in the search. Thus, the range of the raw score is practically not pre-determined. Therefore we normalized it to compute the p-value for each GO term individually and further computed the expected accuracy by examining correlation between the p-value and the accuracy. On the other hand, ESG computes probability values varying between 0 and 1, which can be used directly for comparison and setting cutoffs.

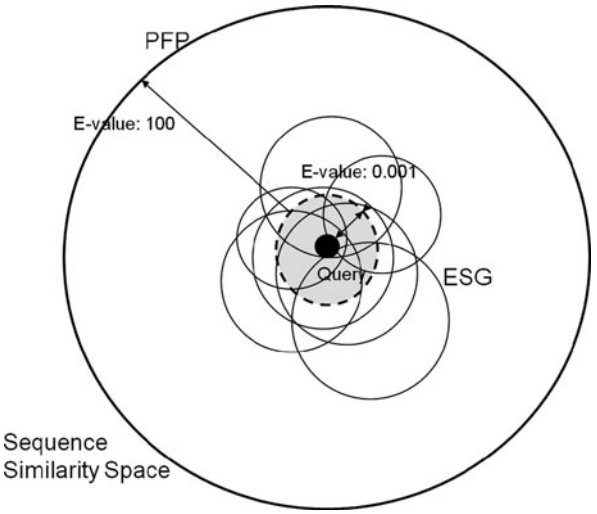


**Fig. 4** Conceptual difference of PFP and ESG

### *PFP and ESG Web Server*

Both PFP and ESG are available as web servers at http://www.kiharalab.org/software.php. Figure 5 shows the job submission page of the ESG web server. Users can enter FASTA format sequences in the text box or upload a file containing one or more amino acid sequences. The parameter "number of hits per stage" indicates the number of PSI-BLAST hits considered at each stage of the ESG algorithm ($N$ in Eq. (7) and $N_i$ in Eq. (8)). Another parameter "number of stages" indicates the number of levels considered, e.g. we considered two levels in the previous section. On the right side panel tutorials are provided for PFP and ESG which explain in detail how to format input and how to download and interpret the results. The web servers also provide ability to create a login account for users which can be used for maintaining users' private jobs and also for checking job status or access results from old jobs. Users can choose to be informed about job completion by receiving an email update.

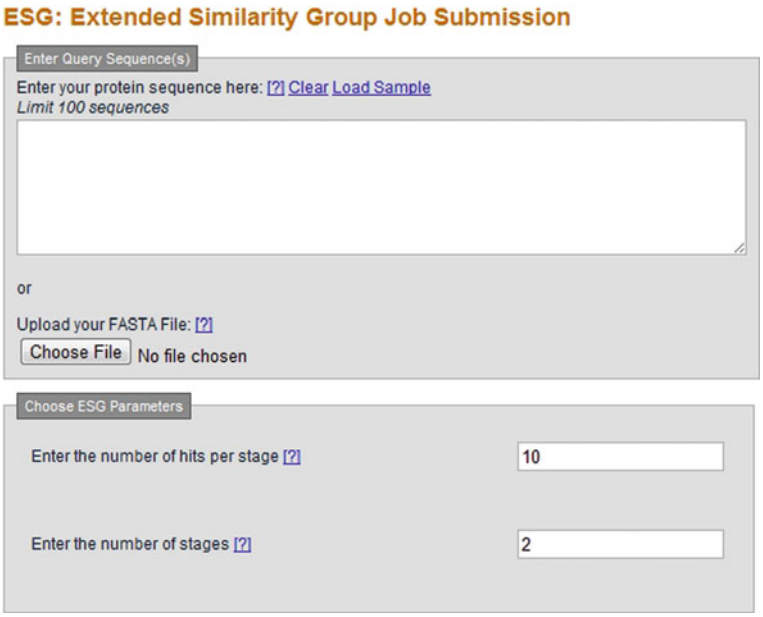


**Fig. 5** ESG web server's job submission page at http://kiharalab.org/esg.php

## Structure of the Gene Functional Space

In the previous sections we have discussed that PFP can significantly increase the annotation coverage of genomes. The larger annotation coverage can benefit biological research in two ways: obviously, functional clues are provided to a larger

number of individual genes. Secondly, we can obtain an overview of the organization of functional space occupied by genomes. And we can further investigate relationship between functions and other important properties of genes, proteins, genomes, and organisms, such as the tertiary structure of proteins, pathways, and gene location in a genome.

To enhance our understanding of the structure of gene functional space, we introduced *functional similarity networks* [35]. We used three genomes, *Escherichia coli* (4,381), *Saccaromyces cerevisiae* (yeast) (6,690), and *Plasmodium falciparum* (malaria) (5,270) for this study. The number of protein genes is shown in the parentheses. *E.coli* and *S. cerevisiae* are well studied model organisms, where over 83.2 and 82.2% of genes, respectively, have been annotated with at least a GO term in the database. *P. falciparum* is an example of less annotated genomes, where only 41.9% of genes have annotation. To the all three genomes, PFP provided a significant number of high confidence predictions, increasing the annotation coverage to 95.2, 96.1, and 90.8%, respectively for *E. coli*, yeast, and the malaria genome.

Using annotated GO terms both in the database and those assigned by PFP, we represented functional similarity of genes in each genome as a network, where genes of similar function are connected with edges. The similarity of sets of GO terms from two genes are quantified using Eq. (11) in chapter "Computational Protein Function Prediction: Framework and Challenges", which compares GO terms in the three categories separately, and also by the three-category version of the *funSim* score (Eq. (12) in chapter "Computational Protein Function Prediction: Framework and Challenges"). Thus, four functional similarity networks, BP-score, MF-score, CC-score, and *funSim*-score networks are computed for each genome (Fig. 6). In all of the functional similarity networks, a majority of the genes are included in the largest connected component.

Analyses of the network properties in comparison with protein-protein interaction networks revealed interesting characteristics of the functional similarity networks. Three parameters of network structures were examined. First, we examined the degree distribution of the networks. The degree distribution concerns the

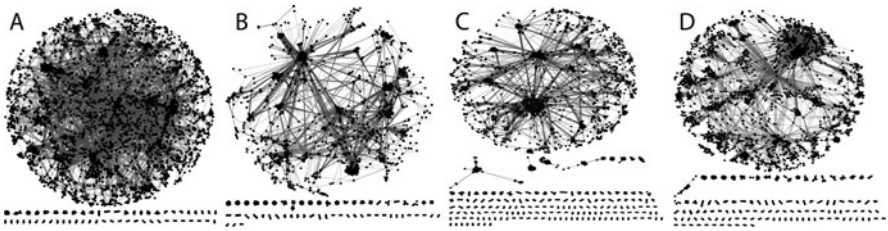


**Fig. 6** Functional similarity networks of yeast genome. (**a**) Similarity of biological process terms in the Gene Ontology are used; (**b**) cellular component terms; (**c**) molecular function terms; (**d**) funSim score is used to define functional similarity. (This figure was modified from fig. 3 in Hawkins et al. [35])

probability of nodes with each number of degree $k$ (edges or connections). If the degree distribution follows the power-law, i.e. $P(k) \sim k^{-\gamma}$, where $\gamma$ is around 1.0, it indicates that the network has few nodes with a large number of connections while the majority of nodes have a small number of connections. In the case of yeast functional similarity networks (Fig. 6), all of them showed a $\gamma$ value close to 1.0, namely, 1.22, 0.83, 0.96, and 1.31, for the BP-score, CC-score, MF-score, and funSim score networks, respectively. It is known that protein-protein interaction (PPI) networks follow the power-law [36]. Indeed, the yeast PPI network has the $\gamma$ value of 1.80. Thus, in general both PPI and the functional similarity networks follow the power-law.

Next examined was the clustering coefficient of the networks. The clustering coefficient of a node indicates how well nodes neighboring to the central nodes are connected to each other. It is defined in the following way:

$$C = \frac{n}{\frac{k(k-1)}{2}} \tag{12}$$

$k$ is the number of neighboring nodes connected to the central node and $n$ is the number of pairs of the neighboring nodes that are directly connected. We consider that a network has high *modularity* if it has a large average clustering coefficient [36, 37]. It turned out that the functional similarity networks distinguish themselves from the PPI networks by having higher clustering coefficient, thus they are highly modular compared to the PPI networks. The clustering coefficient value for the yeast PPI network is 0.10, while the BP-, CC-, MF-, and funSim-score networks showed values of 0.63, 0.77, 0.72, and 0.46, respectively.

We also discussed the network hierarchy based on the network model by Ravasz et al. [37]. A network is considered to be hierarchical if the clustering coefficient, C(k) follows the scaling law, $C(k) \sim k^{-1}$. The clustering degree exponent value $\beta$, $C(k) \sim k^{-1}$ obtained for the functional similarity networks revealed that only the funSim score network has a $\beta$ value close to 1: 0.11, –0.05, 0.40, and 1.39, for the BP-score, CC-score, MF-score, and funSim score networks, respectively. Therefore, interestingly, hierarchy is observed in funSim network (Fig. 7) but not in individual GO-score networks. The network hierarchy was first observed in metabolic pathways [37]. It is an interesting observation that hierarchy of the network arises for the funSim score that integrates single GO-scores, which do not show hierarchy individually. This might imply that the funSim score somewhat captures properties of metabolic pathway networks.

In summary, we studied the landscape of the functional space of genes as the functional similarity networks. Analysis of topological properties of these networks revealed different network properties as compared with the PPI networks. This analysis demonstrates that applying annotations by PFP can have a significant impact in investigating biological systems in an omics scale.

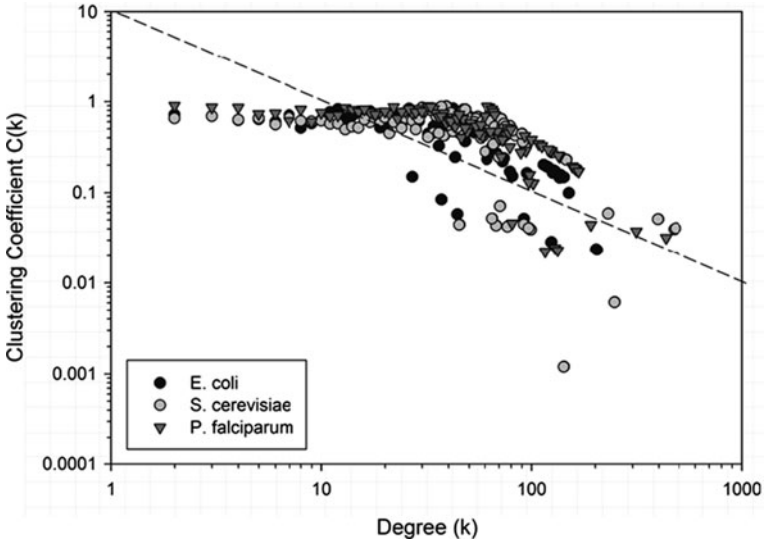


**Fig. 7** Hierarchical modularity of funSim score networks of the three organisms. Clustering coefficient is plotted relative to the degree (k) of nodes. The *dotted lines* shows $C(k) \sim k^{-1}$. (This figure is modified from fig. 5 in Hawkins et al. [35])

## Summary

In this chapter, we introduced two sequence-based function prediction methods developed in our group, PFP and ESG. In contrast to conventional sequence-based function prediction methods, the two methods effectively capture function information in weakly similar sequences. Biological implication by the success of PFP and ESG is that there exist functional commonalities among genes with are not traditionally considered as homologous, and such common functions can be captured by making use of very weakly similar sequences. As the number of sequenced genomes is rapidly increasing, there is even stronger need for sensitive and accurate function prediction methods. These two methods show a new direction for function prediction, which is to explore the twilight zone or even lower sequence similarity, rather than sticking with high sequence similarity or conservation.

**Acknowledgements** MC is supported by grants from Purdue Research Foundation and the Showalter Trust. DK also acknowledges a grant from National Institutes of Health (GM075004) and National Science Foundation (DMS800568, EF0850009, IIS0915801).

## References

1. Ofran, Y., et al. Beyond annotation transfer by homology: novel protein-function prediction methods to assist drug discovery. Drug Discov. Today **10**(21): 1475–1482 (2005).
2. Hawkins, T., et al. PFP: automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. Proteins **74**(3): 566–582 (2009).

3. Hawkins, T., Luban, S., Kihara, D. Enhanced automated function prediction using distantly related sequences and contextual association by PFP. Protein Sci. **15**(6): 1550–1556 (2006).
4. Chitale, M., et al. ESG: extended similarity group method for automated protein function prediction. Bioinformatics **25**(14): 1739–1745 (2009).
5. Altschul, S.F., et al. Basic local alignment search tool. J. Mol. Biol. **215**(3): 403–410 (1990).
6. Pearson, W.R. Rapid and sensitive sequence comparison with FASTP and FASTA. Methods Enzymol. **183**: 63–98 (1990).
7. Pearson, W.R., Lipman, D.J. Improved tools for biological sequence comparison. Proc. Natl. Acad. Sci. USA **85**(8): 2444–2448 (1988).
8. Smith, T.F., Waterman, M.S. Identification of common molecular subsequences. J. Mol. Biol. **147**(1): 195–197 (1981).
9. Brenner, S.E., Chothia, C., Hubbard, T.J. Assessing sequence comparison methods with reliable structurally identified distant evolutionary relationships. Proc. Natl. Acad. Sci. USA **95**(11): 6073–6078 (1998).
10. Hulsen, T., et al. Testing statistical significance scores of sequence comparison methods with structure similarity. BMC Bioinformatics **7**: 444 (2006).
11. Altschul, S.F., et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. **25**(17): 3389–3402 (1997).
12. Pietrokovski, S., Henikoff, J.G. Henikoff, S. The Blocks database – a system for protein classification. Nucleic Acids Res. **24**(1): 197–200 (1996).
13. Hunter, S., et al. InterPro: the integrative protein signature database. Nucleic Acids Res. **37**(Database issue): D211–215 (2009).
14. Finn, R.D., et al. Pfam: clans, web tools and services. Nucleic Acids Res. **34**(Database issue): D247–251 (2006).
15. Attwood, T.K., et al. PRINTS and its automatic supplement, prePRINTS. Nucleic Acids Res. **31**(1): 400–402 (2003).
16. Bru, C., et al. The ProDom database of protein domain families: more emphasis on 3D. Nucleic Acids Res. **33**(Database issue): D212–215 (2005).
17. Hulo, N., et al. The 20 years of PROSITE. Nucleic Acids Res. **36**(Database issue): D245–249 (2008).
18. Letunic, I., et al. SMART 4.0: towards genomic data integration. Nucleic Acids Res. **32**(Database issue): D142–144 (2004).
19. Wilson, D., et al. The SUPERFAMILY database in 2007: families and functions. Nucleic Acids Res. **35**(Database issue): D308–313 (2007).
20. Haft, D.H., Selengut, J.D., White, O. The TIGRFAMs database of protein families. Nucleic Acids Res. **31**(1): 371–373 (2003).
21. Hawkins, T., Chitale, M., Kihara, D. New paradigm in protein function prediction for large scale omics analysis. Mol. Biosyst. **4**(3): 223–231 (2008).
22. Chitale, M., Hawkins, T., Kihara, D. Automated prediction of protein function from sequence. *Prediction of protein strucutre, functions, and interactions*. Bujnicki, J.M. (ed.). New York, NY: Wiley, pp. 63–86 (2009).
23. Kaminska, K.H., Milanowska, K., Bujnicki, J.M. The basics of protein sequence analysis. *Prediction of protein structures, functions, and interactions*. Bujnicki, J.M. (ed.). New York, NY: Wiley, pp. 1–38 (2009).
24. Hennig, S., Groth, D., Lehrach, H. Automated Gene Ontology annotation for anonymous sequence data. Nucleic Acids Res. **31**(13): 3712–3715 (2003).
25. Zehetner, G. OntoBlast function: from sequence similarities directly to potential functional annotations by ontology terms. Nucleic Acids Res. **31**(13): 3799–3803 (2003).
26. Khan, S., et al. GoFigure: automated Gene Ontology annotation. Bioinformatics **19**(18): 2484–2485 (2003).
27. Martin, D.M., Berriman, M., Barton, G.J. GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. BMC Bioinformatics **5**: 178 (2004).
28. Vinayagam, A., et al. GOPET: a tool for automated predictions of Gene Ontology terms. BMC Bioinformatics **7**: 161 (2006).

29. Wass, M.N., Sternberg, M.J. ConFunc – functional annotation in the twilight zone. Bioinformatics **24**(6): 798–806 (2008).
30. Barrell, D., et al. The GOA database in 2009 – an integrated Gene Ontology Annotation resource. Nucleic Acids Res. **37**(Database issue): D396–403 (2009).
31. Gattiker, A., et al. Automated annotation of microbial proteomes in SWISS-PROT. Comput. Biol. Chem. **27**(1): 49–58 (2003).
32. The Universal Protein Resource (UniProt) in 2010. Nucleic Acids Res. **38**(Database issue): D142–148.
33. Zdobnov, E.M., Apweiler, R. InterProScan – an integration platform for the signature-recognition methods in InterPro. Bioinformatics **17**(9): 847–848 (2001).
34. Schlicker, A., et al. A new measure for functional similarity of gene products based on Gene Ontology. BMC Bioinformatics **7**: 302 (2006).
35. Hawkins, T., Chitale, M., Kihara, D. Functional enrichment analyses and construction of functional similarity networks with high confidence function prediction by PFP. BMC Bioinformatics **11**: 265 (2010).
36. Barabasi, A.L., Oltvai, Z.N. Network biology: understanding the cell′s functional organization. Nat. Rev. Genet. **5**(2): 101–13 (2004).
37. Ravasz, E., et al. Hierarchical organization of modularity in metabolic networks. Science **297**(5586): 1551–5 (2002).