

---

# Protein Function Prediction in Proteomics Era

Daisuke Kihara, Troy Hawkins, Stan Luban, Bin Li,  
Karthik Ramani, and Manish Agrawal

**Summary.** The increasing number of genome sequences has become an essential data for biology of this century and function annotation to genes in those genomes is basis of the all biological research. To overcome the limitation of conventional homology-based function annotation methods, here we introduce two types of approaches: A sequence-based approach and a protein surface tertiary shape based approach. The structure-based approach is aimed to predict function of proteins whose tertiary structure was solved. The need of predicting function of proteins from their tertiary structure has emerged by the structural genomics projects, which solve an increasing number of protein structures of unknown function.

## 1 PFP: Extended Sequence-Based Approach

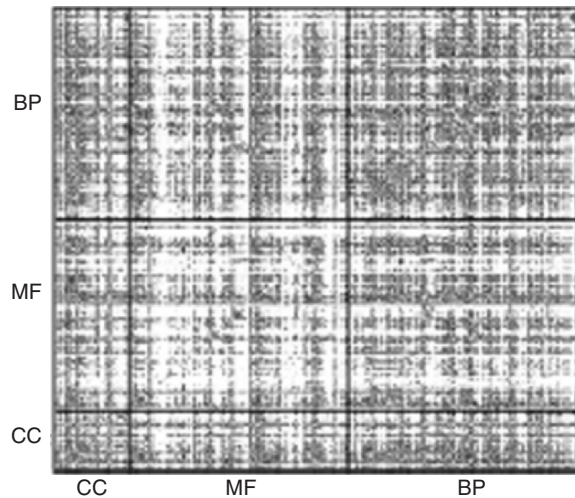
When a new genome is sequenced, one of the first important bioinformatics tasks is function assignment (prediction) to genes in the genome. Usually function of known homologous genes to a new gene found by a database search method, such as BLAST [1], is transferred to the new gene. Different types of sequence-based methods could be also used, which include a Hidden Markov Model database search [2] or a local motif search [3]. One of the shortcomings of these conventional methods is their limited coverage. A BLAST-based annotation typically covers only up to a half of genes in a genome, hence the rest are remained unknown. The small coverage of function in a genome would prevent us from taking full advantage of omics type experiments, e.g., microarray gene expression analysis, protein identification by mass spectrometry analysis, protein-protein interaction analysis. These omics data deduce relationships between genes, but function annotation is crucial to draw biological conclusion from them.

### 1.1 Algorithm of PFP

A typical way to use BLAST is to set up a predefined threshold for the statistical significance score (i.e., *E*-value, a commonly used threshold value is

0.01) and only use highly similar sequences which obtained a score above that threshold value [4]. A more careful way is to identify orthologous genes to a new gene in multiple genomes [5], but again a certain threshold is used for identifying orthologous genes.

Unlike these conventional ways, our novel function prediction methods named Protein Function Prediction (PFP) rather uses even low-scoring sequence hits by a PSI-BLAST search [6] more extensively [7]. PFP uses Gene Ontology (GO) [8] as the vocabulary for gene functions. For an input sequences, PFP outputs ten most probable functions in three GO categories, namely, molecular function (MF), which is essentially biochemical function of proteins, biological process (BP), which indicates pathways where the gene belongs, and cellular component (CC), which specifies localization of the genes in a cell. PFP has the following two key features in the algorithm (1) GO terms associated to all the retrieved sequences by a PSI-BLAST search up to an  $E$ -value of 100 are ranked according to their frequency of occurrence in those sequences and the degree of similarity of the originating sequence to the query and (2) PFP also incorporates function associations, i.e., associations between pairs of GO terms found in UniProt database compiled in the form of the function association matrix (FAM). The FAM describes the frequency at which two GO terms occur together in the same context by quantifying the co-occurrence of each pair of annotations within UniProt sequences. Figure 1 gives a graphical representation of FAM. GO terms in the three functional categories are aligned on the two axis of the matrix, and the association between GO pairs is shown in a gray scale.



**Fig. 1.** UniProt FAM. Associations of GO terms within and across the three categories. Darker spots indicate a higher association score.

The PFP score given to a function  $f_a$  is computed as follows:

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

where  $N$  is the number of sequences retrieved by PSI-BLAST,  $M$  is the number of GO terms associated to the sequence  $i$ ,  $E\_value(i)$  is the  $E$ -value given to the sequence  $i$ ,  $f_j$  is a GO term assigned to the sequence  $i$ ,  $P(f_a|f_j)$  is the conditional probability that  $f_a$  is associated with  $f_j$  computed by the FAM matrix, and  $b$  is a parameter. Following (1), GO terms which repeatedly occur in a PSI-BLAST search will stand out in the PFP score.

## 1.2 Results

We benchmarked PFP on a 2,000 nonredundant sequences randomly selected from UniProt database. In order to examine PFP's ability to mine correct function from low-scoring sequences in a PSI-BLAST search, the most significant hits using several  $E$ -value cutoffs are ignored. We use the default  $E$ -value cutoff ( $-e$  10) and  $E$ -value threshold for inclusion in multiple iterations ( $-h$  0.005), and set the maximum number of iterations to three ( $-j$  3). Figure 2 shows the fraction of test sequences whose biological process (BP) is correctly predicted by PFP in the top five scoring prediction. A prediction is counted as correct if the common parental node of a predicted GO term and the correct GO term is equal to or deeper than the depth of two in the GO hierarchical tree. The  $E$ -value cutoff value ( $x$ -axis) represents the minimum similarity for sequences used in the benchmark analysis: Retrieved sequences with the  $E$ -value cutoff or smaller (more significant) are not used for function prediction by PFP and PSI-BLAST. PSI-BLAST annotations are transferred from

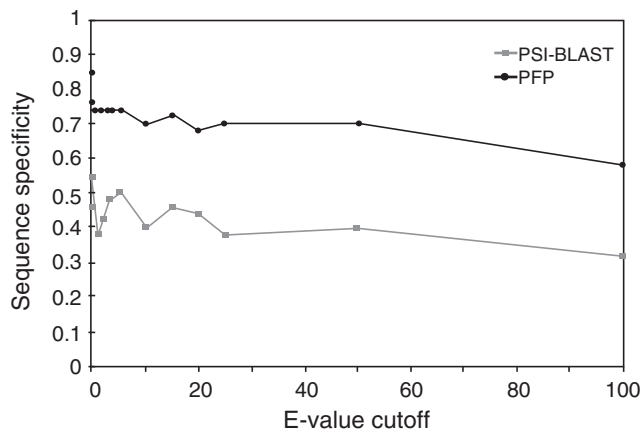


Fig. 2. Sequence-level specificity of PFP and conventional PSI-BLAST

the most similar sequence scoring above each  $E$ -value cutoff. It is evident that PFP outperforms PSI-BLAST in the entire range of the  $E$ -value cutoff with an accuracy which is almost double of that of PSI-BLAST. It is remarkable that the accuracy does not drop much even when only sequences of an  $E$ -value of 10 or larger are used.

PFP was ranked the best in a protein function prediction competition held at the Automated Function Prediction Special Interest Group (AFP-SIG) meeting in the 13th Annual International Conference on Intelligent Systems for Molecular Biology (ISMB) in June, 2005. See the recent paper for our performance at the AFP-SIG meeting and further benchmark results [7]. PFP is publicly available as a Web server at <http://dragon.bio.purdue.edu/pfp/>.

## 2 Toward Tertiary Structure-Based Function Prediction

In this section our protein surface shape-based function prediction method is outlined. Generally, a structure-based function prediction works in two steps (1) identifying potentially functionally important sites on a protein surface; and (2) comparing the identified site to known sites stored in a database. And if a similar site is found in this search, function associated to the known site (e.g., ATP binding) is transferred to the query protein. Each of them is described later.

### 2.1 Identification of Active Sites of Proteins

This step scans the surface of a protein to identify several candidates of active sites characterized by their local landscape, distribution of electrostatic potential, and/or residue conservation among the family. The key feature of a functionally important site is its local landscape, i.e., cavities and protrusions. For enzyme proteins, identifying cavities in a surface is of a special importance because their active sites bind a ligand molecule to catalyze chemical reaction. We have developed a novel algorithm for fast protein pocket region detection by checking *visibility* on the protein surface.

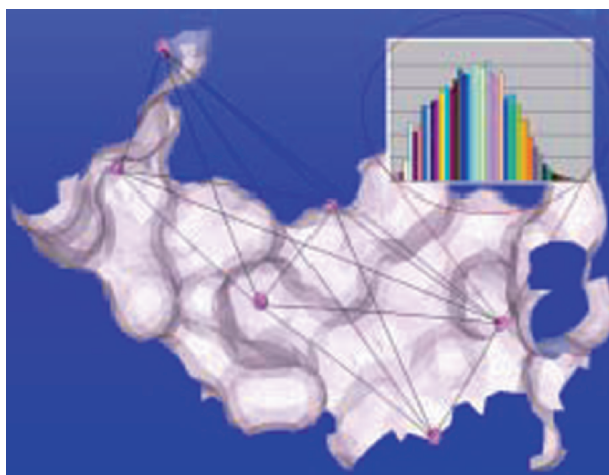
The algorithm called VisGrid first projects a query protein structure on to a three-dimensional grid (voxelization). A cavity is identified as a group of voxels occupied by the protein volume which have less visibility, that is, those which are surrounded by a certain number of empty voxels. The visibility of a voxel on the protein surface is defined as the percentage of viewable directions from the voxel. Voxels at the bottom of a pocket have a low visibility and those at the top of a protrusion have a high visibility. After identifying voxels with a low visibility on the surface, they are clustered into several local sites. Then the identified sites are ranked by their size (i.e., the number of voxels). VisGrid can identify not only pockets on the surface of a protein, but also hollows which are almost buried inside of a protein. This is important because there are often cases where a ligand molecule is held internal core of a protein.

We tested VisGrid on a benchmark dataset used by An et al. [9]. This dataset (LP\_SET) consists of 5,561 protein structures taken from PDB, which include naturally occurring hetero molecules of a reasonable size. VisGrid identifies 79.1% of the active site residues (defined as the residues which locate closer than 4.5 Å to the ligand molecule) as the first rank; 87.6% of the binding site residues within the top three rank; and 91.4% among all the sites predicted (sensitivity). The detail of VisGrid will be published elsewhere.

## 2.2 Comparison of Local Sites of Proteins

Once characteristic local sites are identified in a query protein structure by VisGrid, the next step is to compare the sites with known functionally important sites in proteins. Here representation of local sites and comparison are the crucial steps which are intertwined with each other. We developed a representation of a local site which has following components (1) identifying *critical points* (2) representing the surrounding region of each critical point as *histograms* and (3) representing the entire site surface as a *graph*.

Initially a surface of the local site is represented by triangle tessellae computed by MSMS algorithm [10]. For each vertex of the tessellae, the local curvature is computed using a neighboring area within a certain radius  $R$ . Then vertices which has either the local maximum (i.e., a protrusion) or the local minimum (i.e., a pocket) curvature within the radius  $R$  is identified as *critical points* of the site. The entire local site can be represented as a complete graph (a graph with fully connected nodes) of the critical points. Now the local landscape of a critical point is described by a histogram of distances between all pairs of vertices in the area of the radius  $R$  (Fig. 3).



**Fig. 3.** Local site representation. The entire site is represented as the complete graph of the critical points shown in pink

Hence comparison of two local sites becomes a graph comparison problem. Since we want to identify global as well as local similarity between two sites represented as graphs, finding maximal common regions in the two graphs will be an appropriate measure of similarity. The similarity of two graphs reflects matching of the length of corresponding edges and of corresponding nodes. For matching of nodes, currently only histograms which capture their local geometry information is compared, but physicochemical properties or residue conservation information of critical points will be implemented. Comparison of two graphs comes down to the clique detection problem when an association graph is constructed from the two graphs.

### 3 Concluding Remarks

We have overviewed sequence- and structure-based function prediction methods currently being developed in our group. Rapid developments of new omics experimental techniques and the progress of structural genomics projects have seriously demanded renovation of bioinformatics tools – and here are a part of our responses to meet their needs.

### Acknowledgment

This work was supported in part by the National Institute of General Medical Sciences of the National Institutes of Health (GM-075004), Purdue Research Foundation, and Purdue Alumni Association.

### References

1. Altschul SF et al. (1990) *J Mol Biol* 215:403–410
2. Bateman A et al. (2002) *Nucleic Acids Res* 30:276–280
3. Hulo N et al. (2004) *Nucleic Acids Res* 32:D134–D137
4. Pearson WR (1996) *Methods Enzymol.* 266:227–258
5. Tatusov RL et al. (2003) *BMC Bioinformatics* 2003, 4:41
6. Altschul SF et al. (1997) *Nucleic Acids Res* 25:3389–3402
7. Hawkins T, Luban S, Kihara D (2006) *Protein Sci In press* 15:1550–1556
8. Harris MA et al. (2004) *Nucleic Acids Res* 32:D258–D261
9. An J, Totrov M, Abagyan R (2005) *Mol Cell Proteomics.* 4:752–761
10. Sanner M, Olson AJ, Spehner JC (1995) *Proceedings of 11th ACM Symposium on Computational Geometry C6–C7*