Supplementary Information

"A large-scale evaluation of computational protein function prediction" by Radivojac P, et al. *Nature Methods*, 2013.

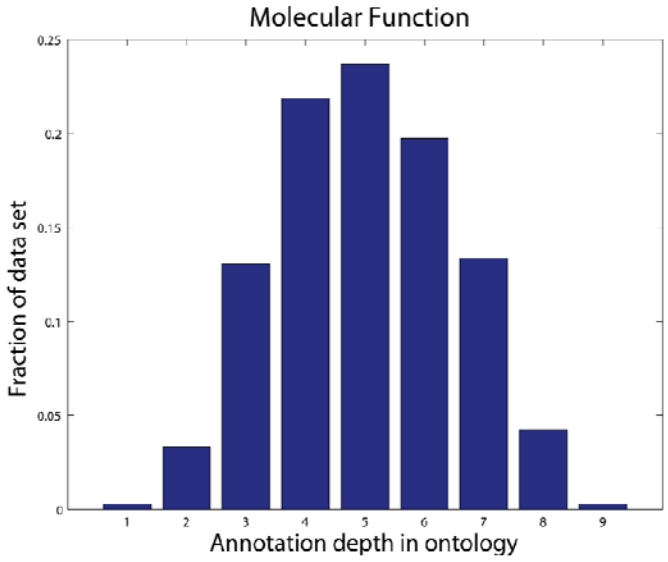
Content:

1. Supplementary Figures 1-8.
2. Supplementary Table 3.
3. Supplementary Note containing short descriptions of all participating methods and detailed description of unpublished methods ranked overall in the top ten.

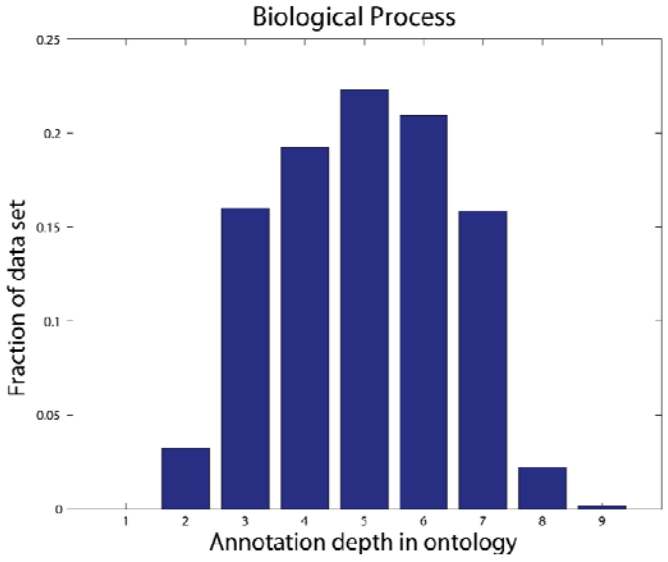
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 1. Distribution of depths of the leaf annotations over all targets in (A) Molecular Function ontology and (B) Biological Process ontology. A leaf term for a target is defined as any term whose descendent nodes (more specific nodes) are not among the experimentally determined terms for that protein.

Supplementary Figure 1A:



Supplementary Figure 1B:

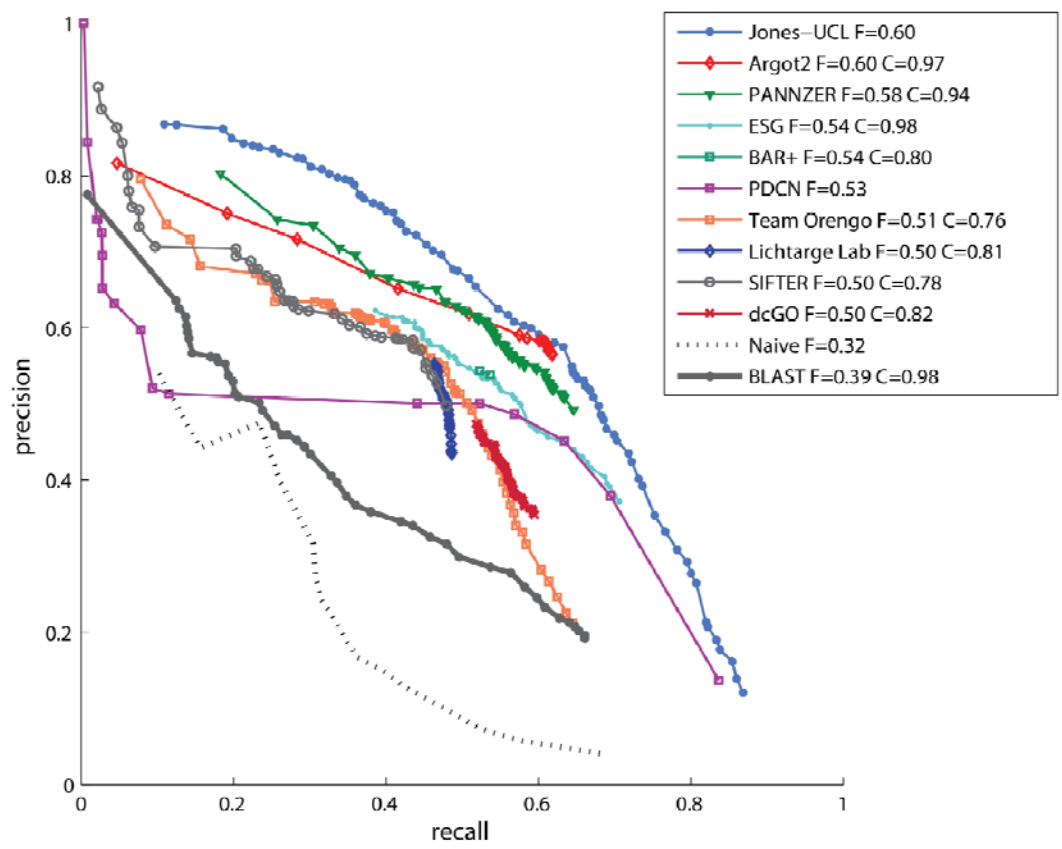


Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 2. Precision-recall curves for the top-performing methods for (A) Molecular Function ontology, excluding proteins annotated with "protein binding" leaf term only, (B) Molecular Function ontology, (C) Biological Process ontology, (D) Biological Process ontology including only GOslim terms. All panels show the top ten participating methods in each category, as well as the BLAST and Naïve baseline methods. The legend provides the maximum F-measure (*F*) for all methods and coverage (*C*) for all methods that did not make predictions on all targets. In cases where a Principal Investigator (PI) participated with multiple teams, only the results of the best scoring method are presented. Previously unpublished methods are denoted as *experimental*. Methods: Jones-UCL[54] (PI: David Jones, University College London), Argot2[55] (PI: Stefano Toppo, University of Padova), PANNZER (PI: Liisa Holm, University of Helsinki; experimental method), ESG[56] (PI: Daisuke Kihara, Purdue University), BAR+[57, 58] (PI: Rita Casadio, University of Bologna), PDCN[59] (PI: Jianlin Cheng, University of Missouri), Team Orengo[60] (PI: Christine Orengo, University College London), Lichtarge Lab[61] (PI: Olivier Lichtarge, Baylor College of Medicine), SIFTER[20, 62] (PI: Steven Brenner, University of California, Berkeley), dcGO[63, 64] (PI: Julian Gough, University of Bristol), MS-kNN[65] (PI: Slobodan Vucetic, Temple University), GOstruct[34] (PI: Asa Ben-Hur, Colorado State University), BMRF[33] (PI: Cajo J. F. ter Braak, Wageningen University), Rost Lab[66] (PI: Burkhard Rost, Technische Universität München), Tian Lab (PI: Weidong Tian, Fudan University; experimental method).

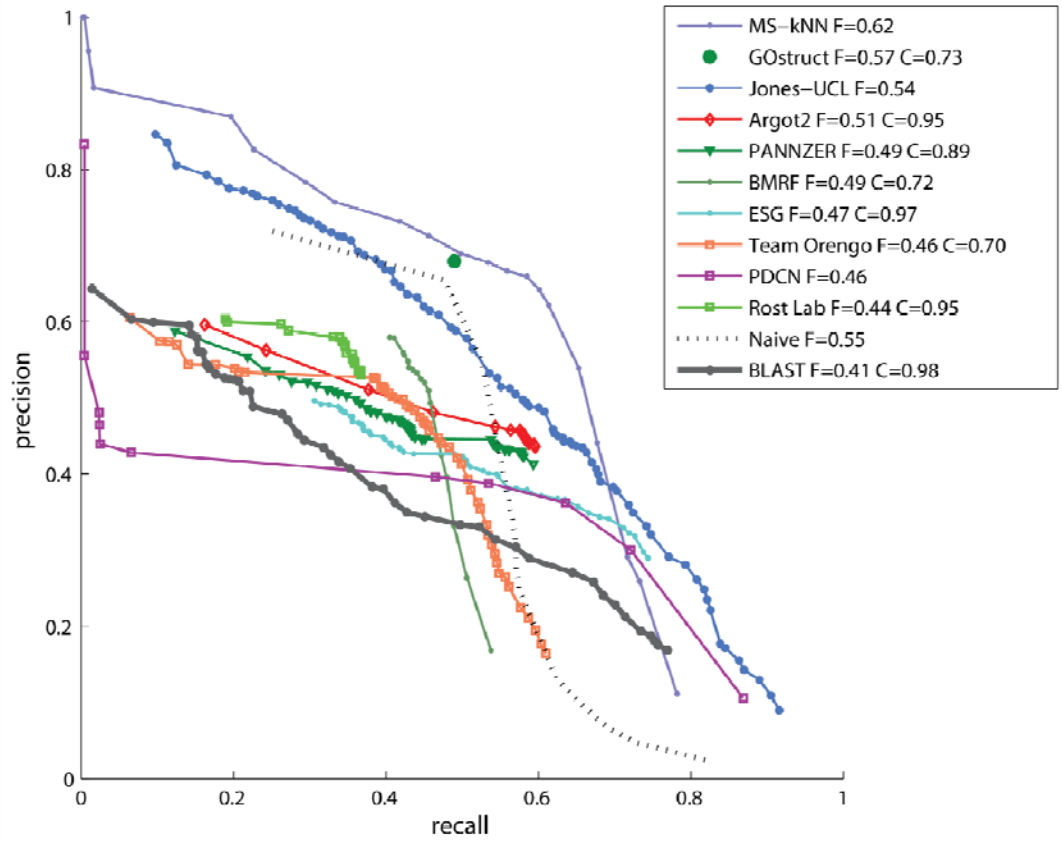
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 2A:



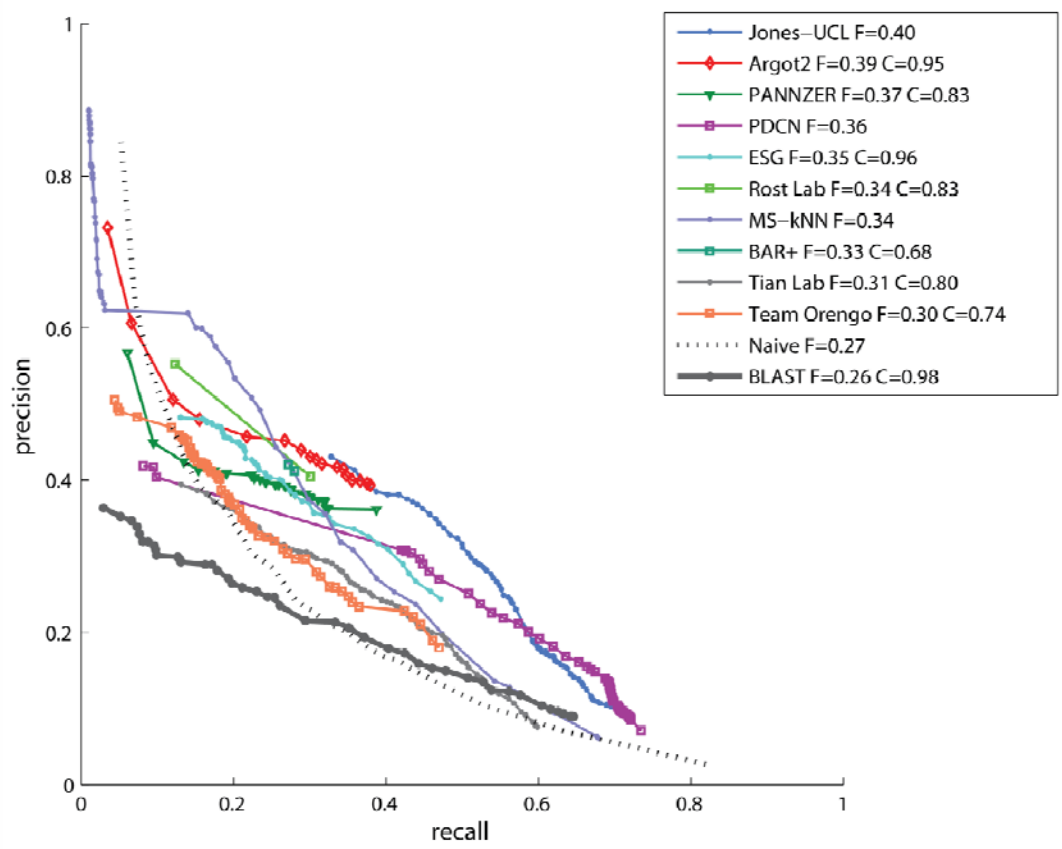
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 2B:



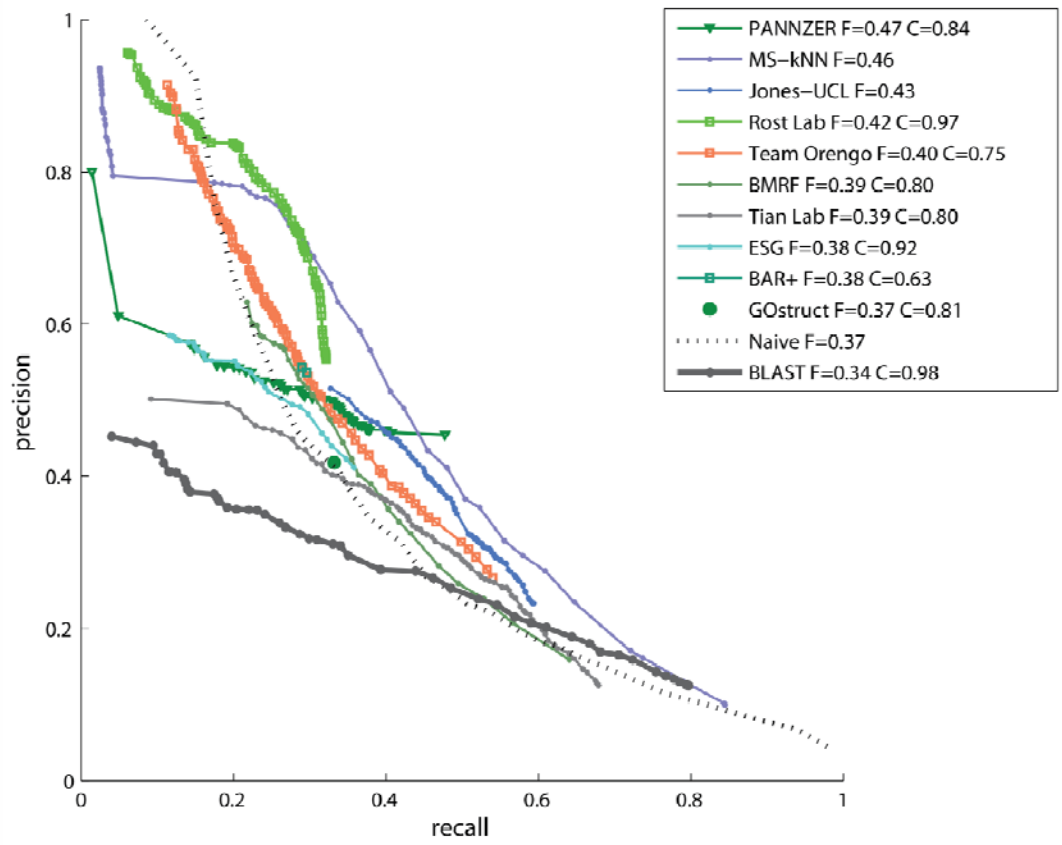
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 2C:



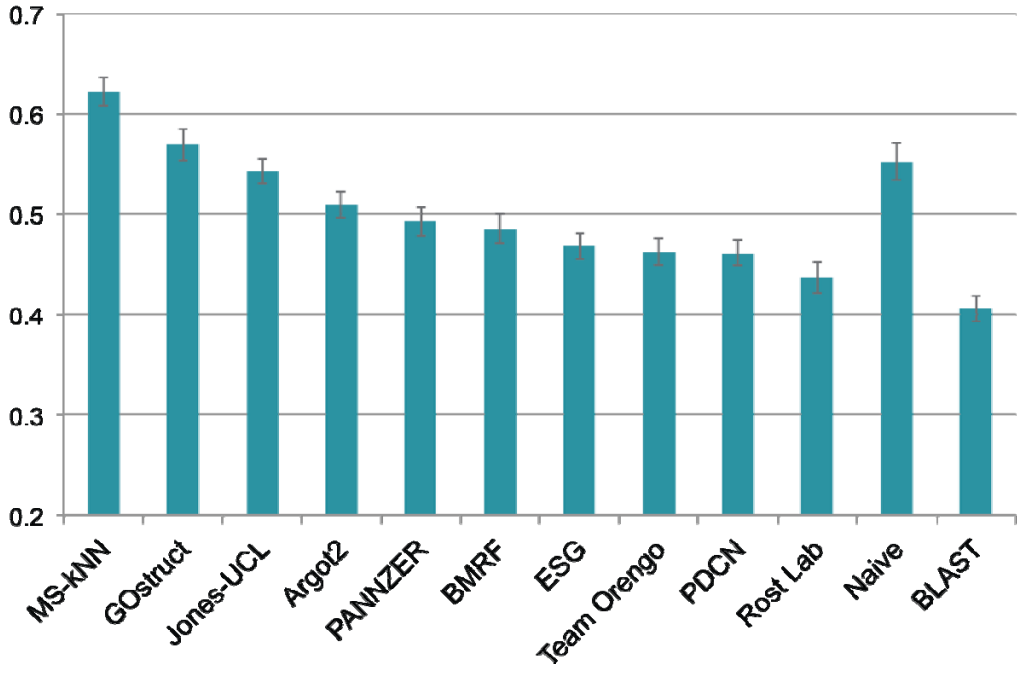
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 2D:



Nature Methods: doi:10.1038/nmeth.2340

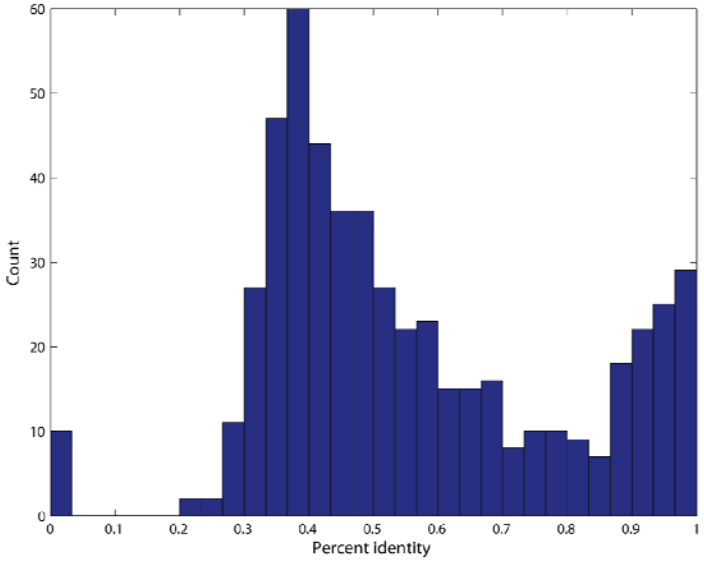
Supplementary Figure 3. Performance evaluation based on the maximum F-measure for the top performing methods for the Molecular Function ontology in cases where all proteins were included in evaluation (Fig. 2 in the manuscript shows equivalent evaluation when proteins characterized with "protein binding" as their only leaf term were excluded). Bars show the top 10 participating methods as well as the BLAST and Naïve baseline methods. A perfect predictor would be characterized with the F_{max} of 1. Confidence intervals (95%) were determined using bootstrapping with 10,000 iterations on the set of target sequences. In cases where a Principal Investigator (PI) participated with multiple teams, only the results of the best-scoring method are presented. Previously unpublished methods are denoted as "experimental". Methods: MS-kNN[65] (PI: Slobodan Vucetic, Temple University), GOstruct[34] (PI: Asa Ben-Hur, Colorado State University), Jones-UCL[54] (PI: David Jones, University College London), Argot2[55] (PI: Stefano Toppo, University of Padova), PANNZER (PI: Liisa Holm, University of Helsinki; experimental method), BMRF[33] (PI: Cajo J. F. ter Braak, Wageningen University), ESG[56] (PI: Daisuke Kihara, Purdue University), Team Orengo[60] (PI: Christine Orengo, University College London), PDCN[59] (PI: Jianlin Cheng, University of Missouri), Rost Lab[66] (PI: Burkhard Rost, Technische Universität München).



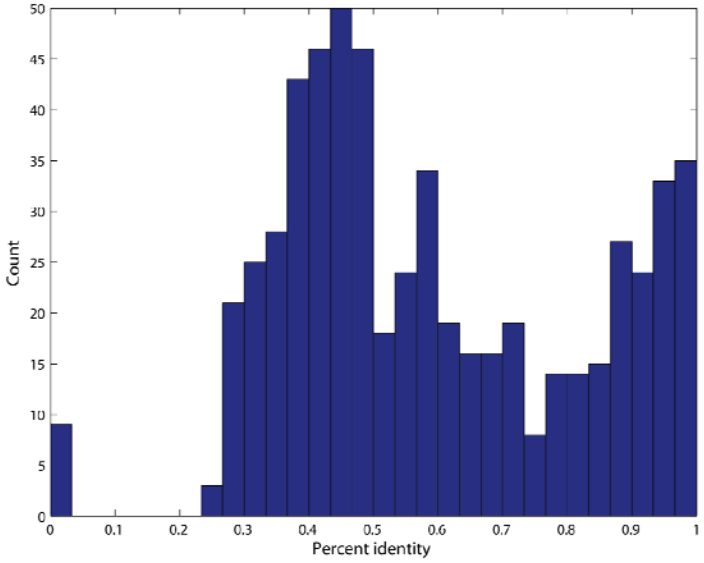
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 4. The histogram of pairwise sequence identities between each target and the experimentally annotated template most similar to it: (A) Molecular Function ontology, and (B) Biological Process ontology. The histograms roughly determine two groups of targets: easy – with maximum sequence identity greater than or equal to 60%, and difficult – with maximum sequence identity below 60%.

Supplementary Figure 4A:



Supplementary Figure 4B:

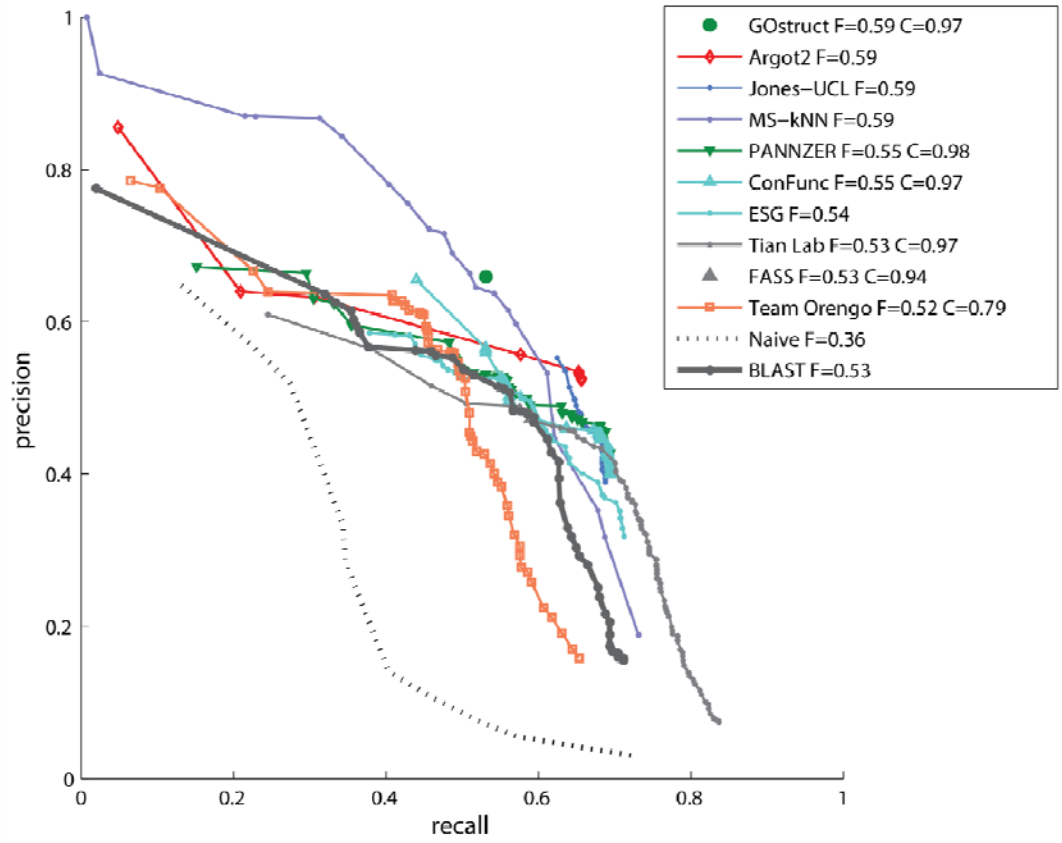


Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 5. Precision-recall curves for the top-performing methods for: (A) easy targets and Molecular Function ontology without "protein binding"; (B) easy targets and Molecular Function ontology; (C) difficult targets and Molecular Function ontology, without "protein binding"; (D) difficult targets and Molecular Function ontology; (E) easy targets and Biological Process ontology; (F) difficult targets and Biological Process ontology. All panels show the top ten participating methods in each category, as well as the BLAST and Naïve baseline methods. The legend provides the maximum F-measure (*F*) for all methods and coverage (*C*) for all methods that did not make predictions on all targets. In cases where a Principal Investigator (PI) participated with multiple teams, only the results of the best-scoring method are presented. Previously unpublished methods are denoted as "experimental". Methods: GOstruct[34] (PI: Asa Ben-Hur, Colorado State University), Argot2[55] (PI: Stefano Toppo, University of Padova), Jones-UCL[54] (PI: David Jones, University College London), MS-kNN[65] (PI: Slobodan Vucetic, Temple University), PANNZER (PI: Liisa Holm, University of Helsinki; experimental method), ConFunc[13] (PI: Michael Sternberg, Imperial College; experimental method), ESG[56] (PI: Daisuke Kihara, Purdue University), Tian Lab (PI: Weidong Tian, Fudan University; experimental method), FASS (PI: Rajendra Joshi, Centre for Development of Advanced Computing; experimental method), Team Orengo[60] (PI: Christine Orengo, University College London), BMRF[33] (PI: Cajo J. F. ter Braak, Wageningen University), BAR+[57, 58] (PI: Rita Casadio, University of Bologna), PDCN[59] (PI: Jianlin Cheng, University of Missouri), SIFTER[20, 62] (PI: Steven Brenner, University of California, Berkeley), Lichtarge Lab[61] (PI: Olivier Lichtarge, Baylor College of Medicine), dcGO[63, 64] (PI: Julian Gough, University of Bristol), Rost Lab[66] (PI: Burkhard Rost, Technische Universität München).

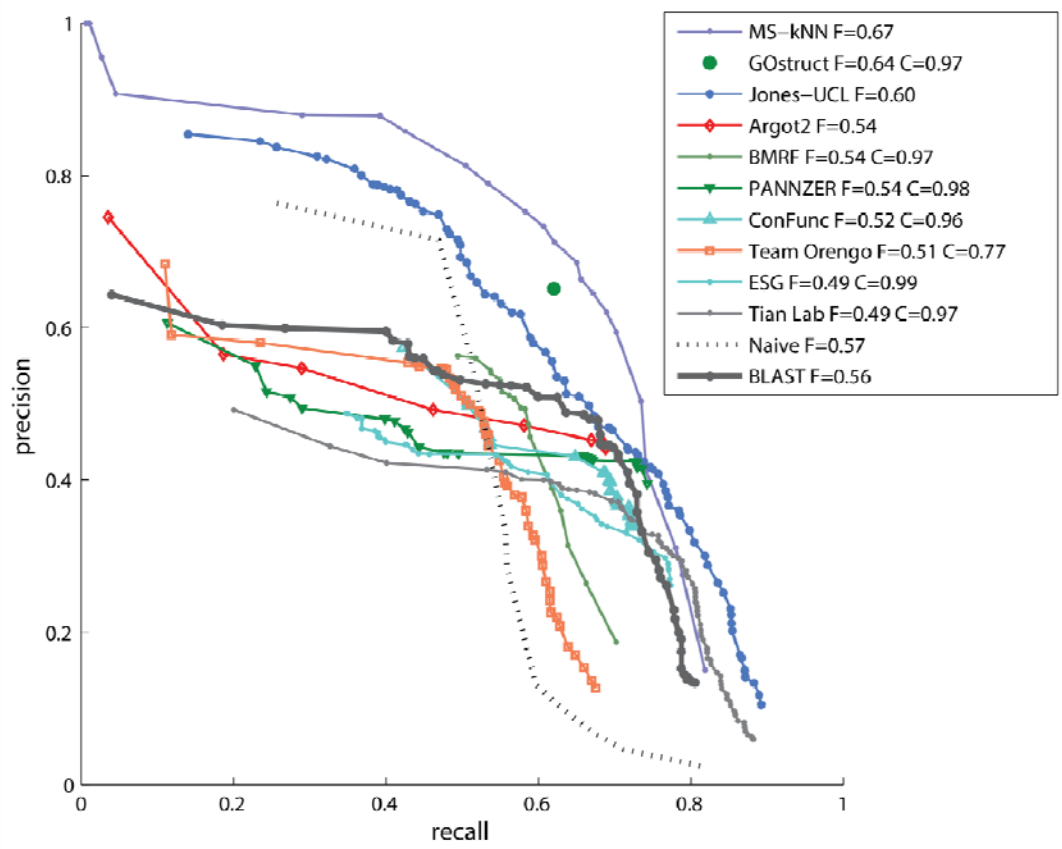
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 5A:



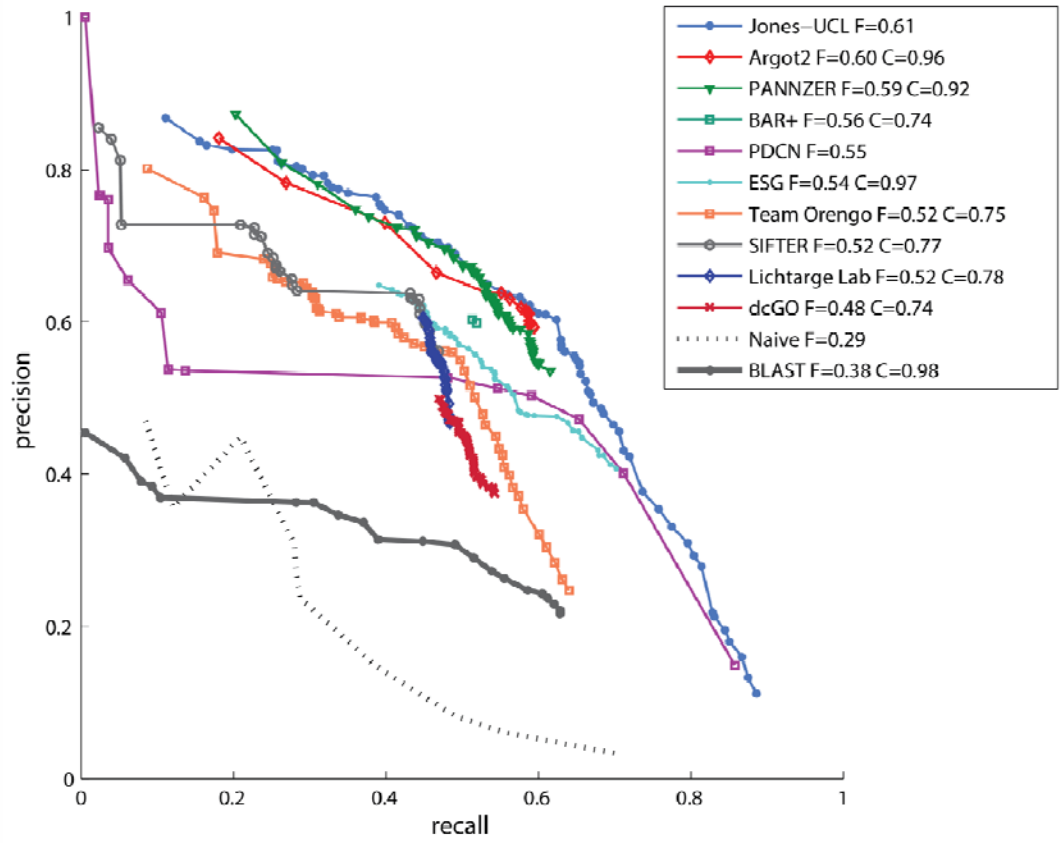
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 5B:



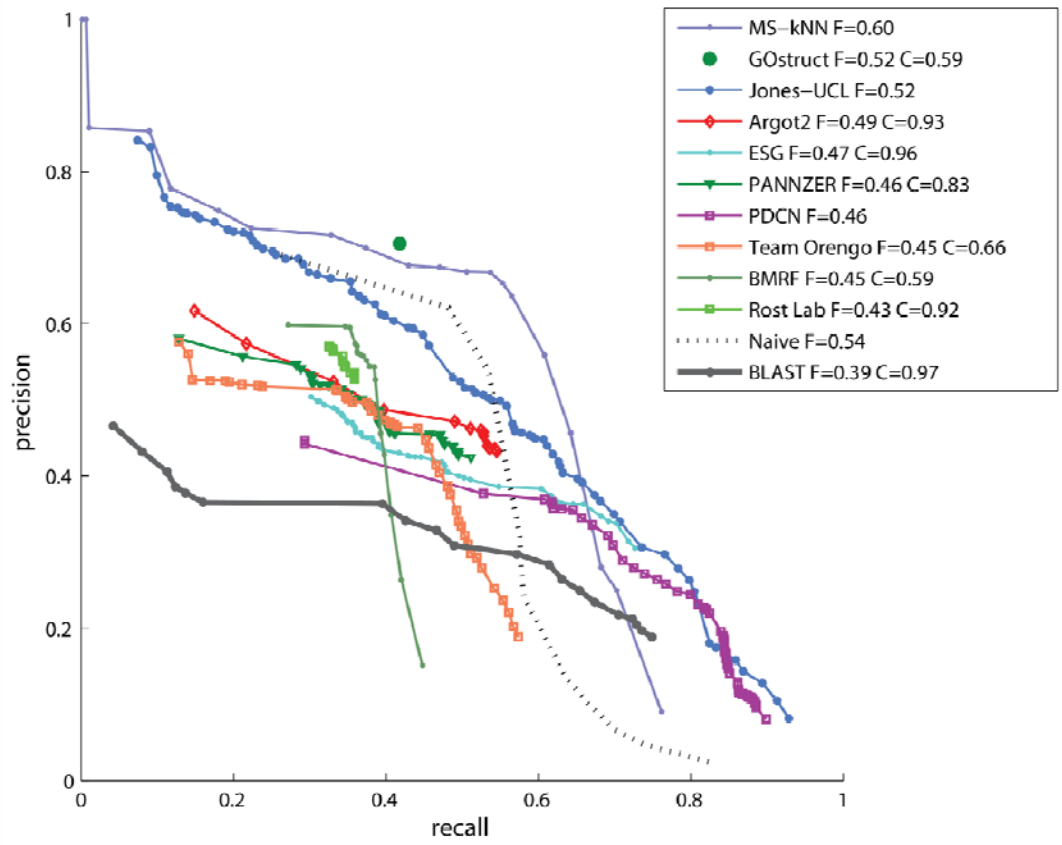
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 5C:



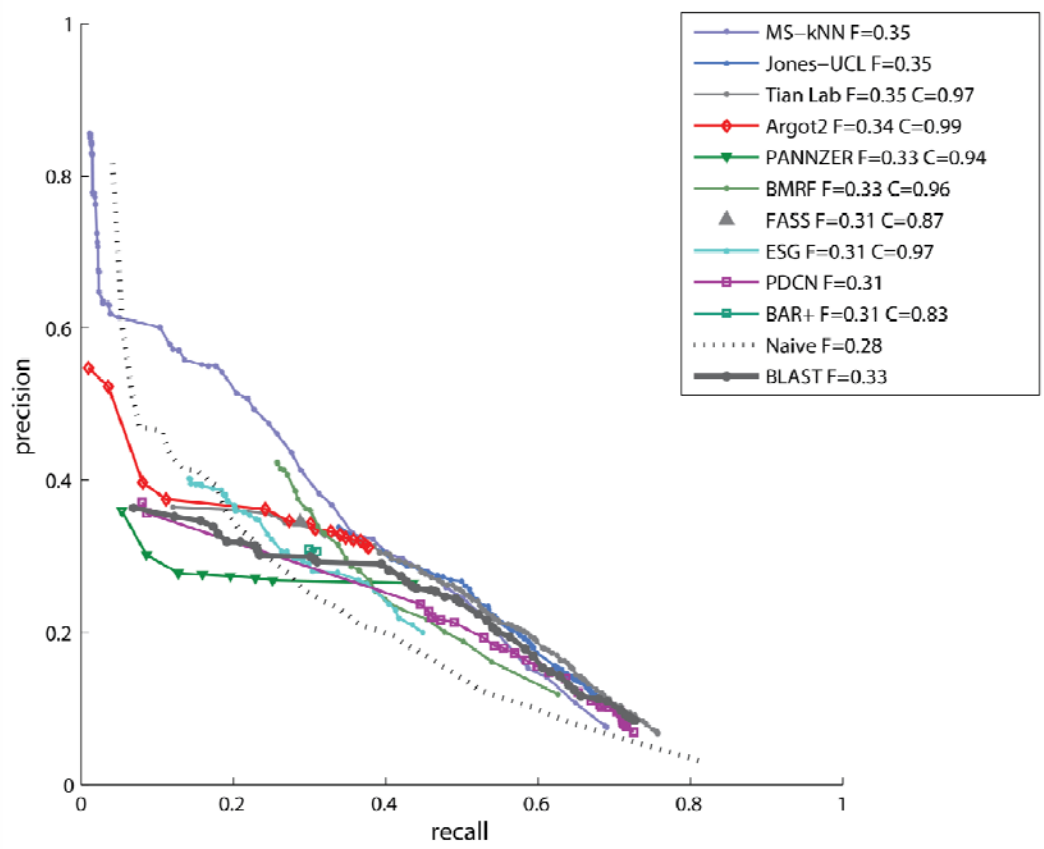
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 5D:



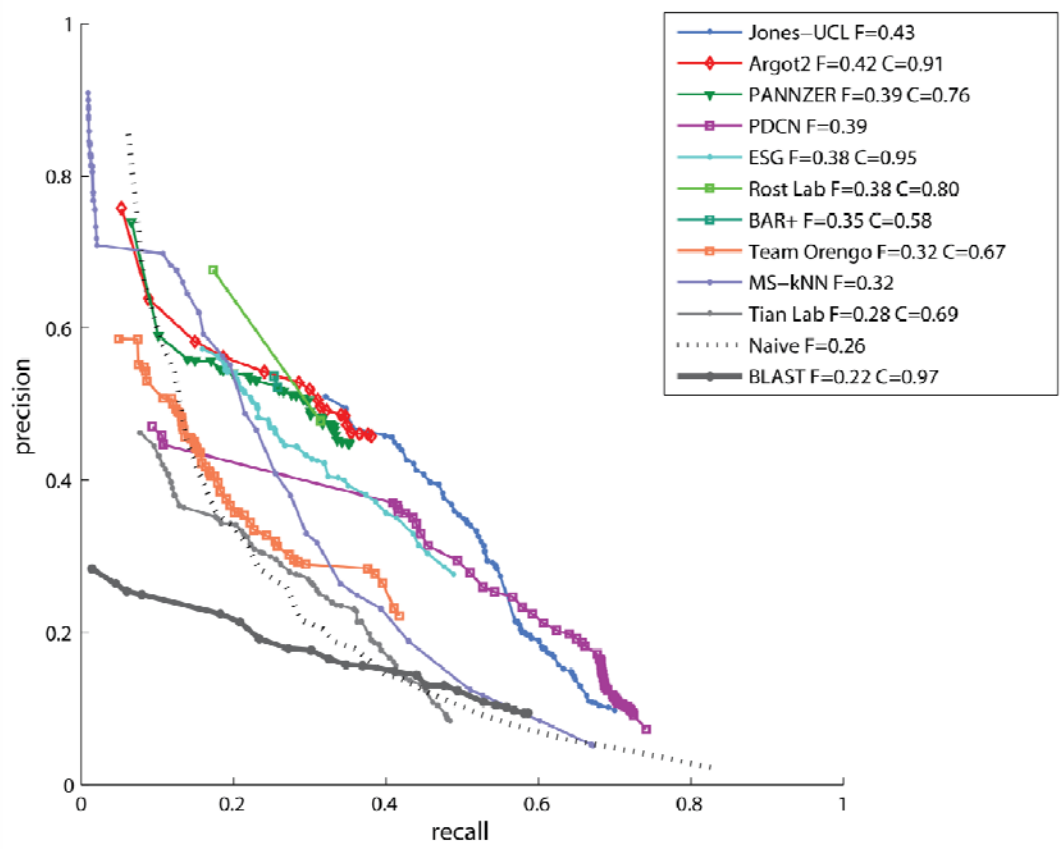
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 5E:



Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 5F:

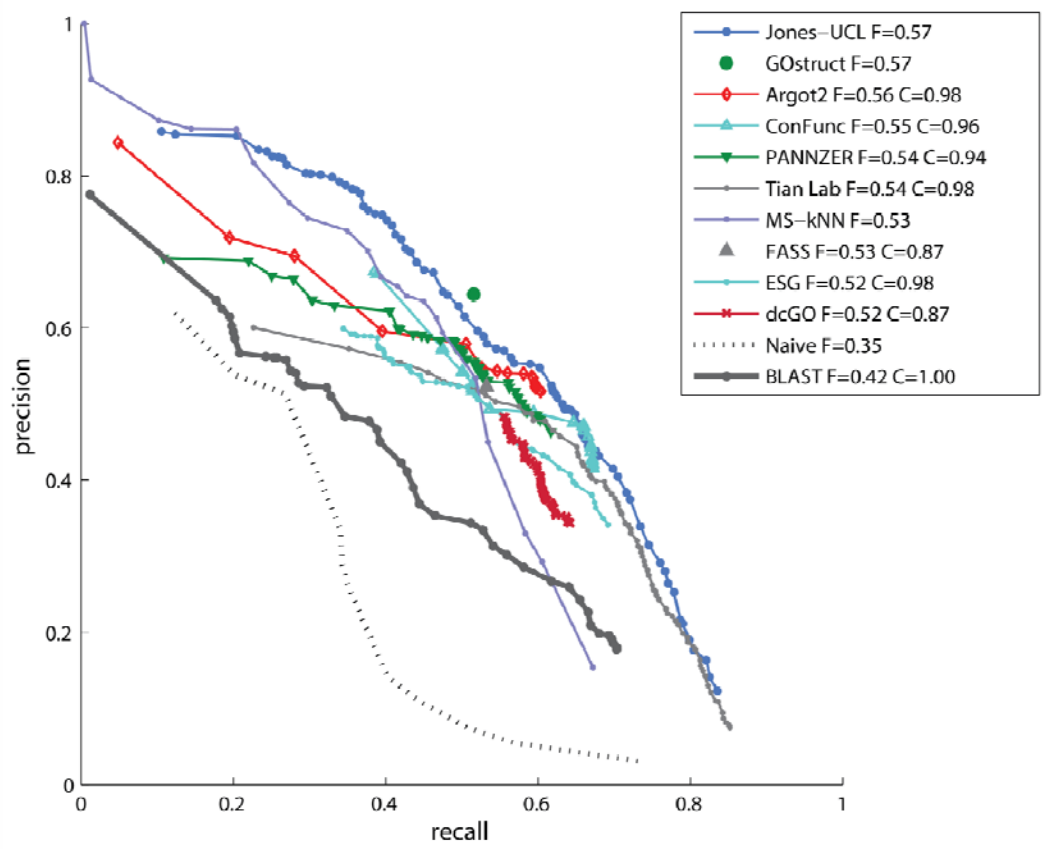


Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 6: Precision-recall curves for the top-performing methods for (A) eukaryotic targets and Molecular Function ontology without "protein binding"; (B) eukaryotic targets and Molecular Function ontology; (C) prokaryotic targets and Molecular Function ontology without "protein binding"; (D) prokaryotic targets and Molecular Function ontology; (E) eukaryotic targets and Biological Process ontology; (F) prokaryotic targets and Biological Process ontology. All panels show the top ten participating methods in each category, as well as the BLAST and Naïve baseline methods. The legend provides the maximum F-measure (*F*) for all methods and coverage (*C*) for all methods that did not make predictions on all targets. In cases where a Principal Investigator (PI) participated with multiple teams, only the results of the best scoring method are presented. Previously unpublished methods are denoted as *experimental*. Methods: Jones-UCL[54] (PI: David Jones, University College London), GOstruct[34] (PI: Asa Ben-Hur, Colorado State University), Argot2[55] (PI: Stefano Toppo, University of Padova), ConFunc[13] (PI: Michael Sternberg, Imperial College; experimental method), PANNZER (PI: Liisa Holm, University of Helsinki; experimental method), Tian Lab (PI: Weidong Tian, Fudan University; experimental method), MS-kNN[65] (PI: Slobodan Vucetic, Temple University), FASS (PI: Rajendra Joshi, Centre for Development of Advanced Computing; experimental method), ESG[56] (PI: Daisuke Kihara, Purdue University), dcGO[63, 64] (PI: Julian Gough, University of Bristol), BMRF[33] (PI: Cajo J. F. ter Braak, Wageningen University), Rost Lab[66] (PI: Burkhard Rost, Technische Universität München), Team Orengo[60] (PI: Christine Orengo, University College London), BAR+[57, 58] (PI: Rita Casadio, University of Bologna), PDCN[59] (PI: Jianlin Cheng, University of Missouri), Lichtarge Lab[61] (PI: Olivier Lichtarge, Baylor College of Medicine), SIFTER[20, 62] (PI: Steven Brenner, University of California, Berkeley), GORBI (PI: Tomislav Šmuc, Ruđer Bošković Institute).

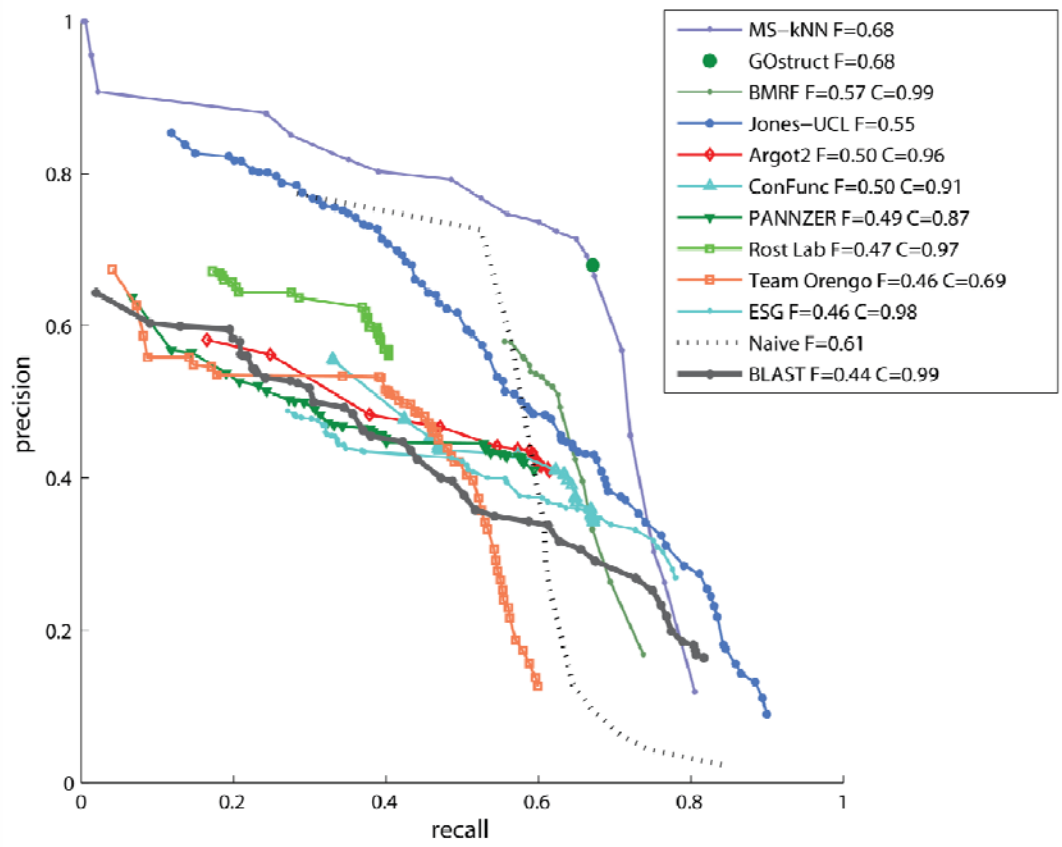
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 6A:



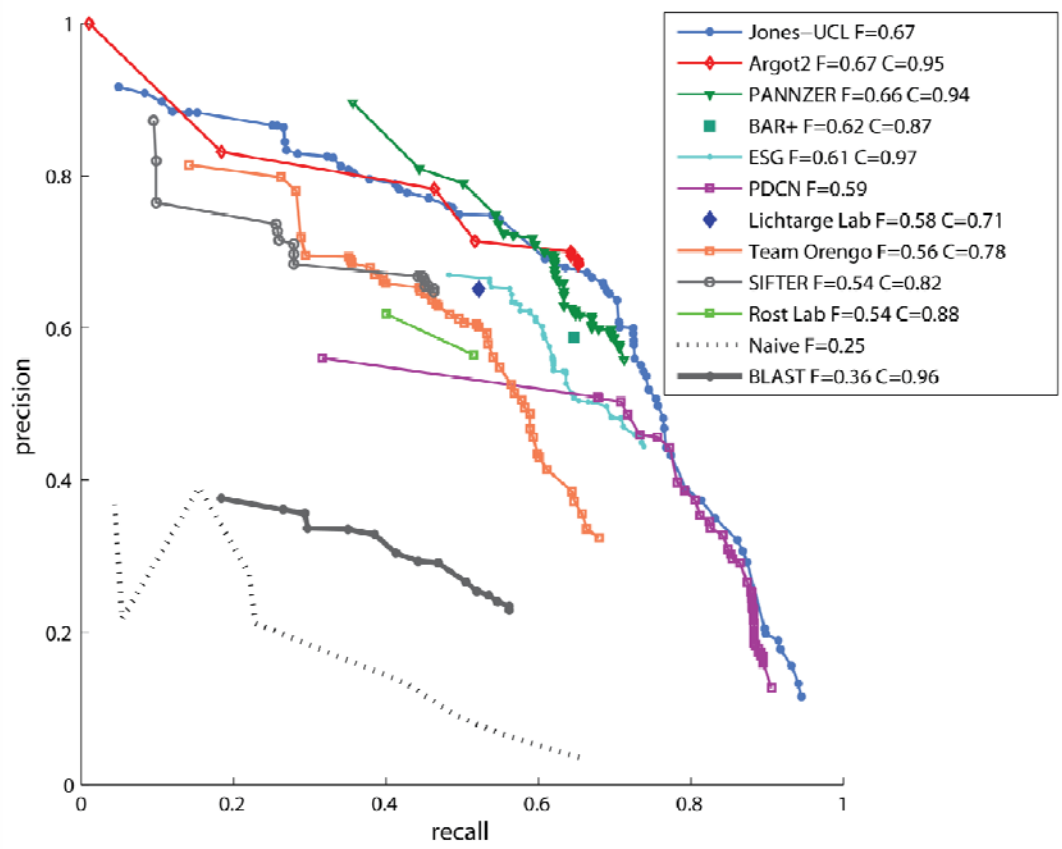
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 6B:



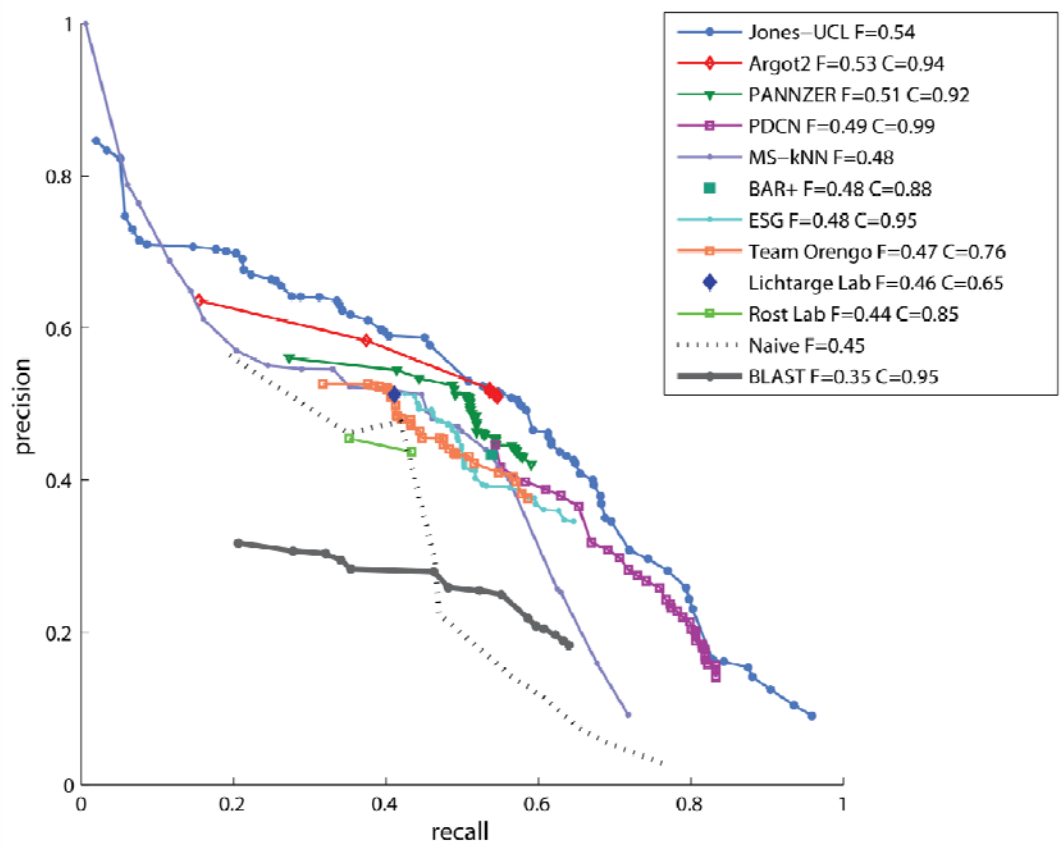
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 6C:



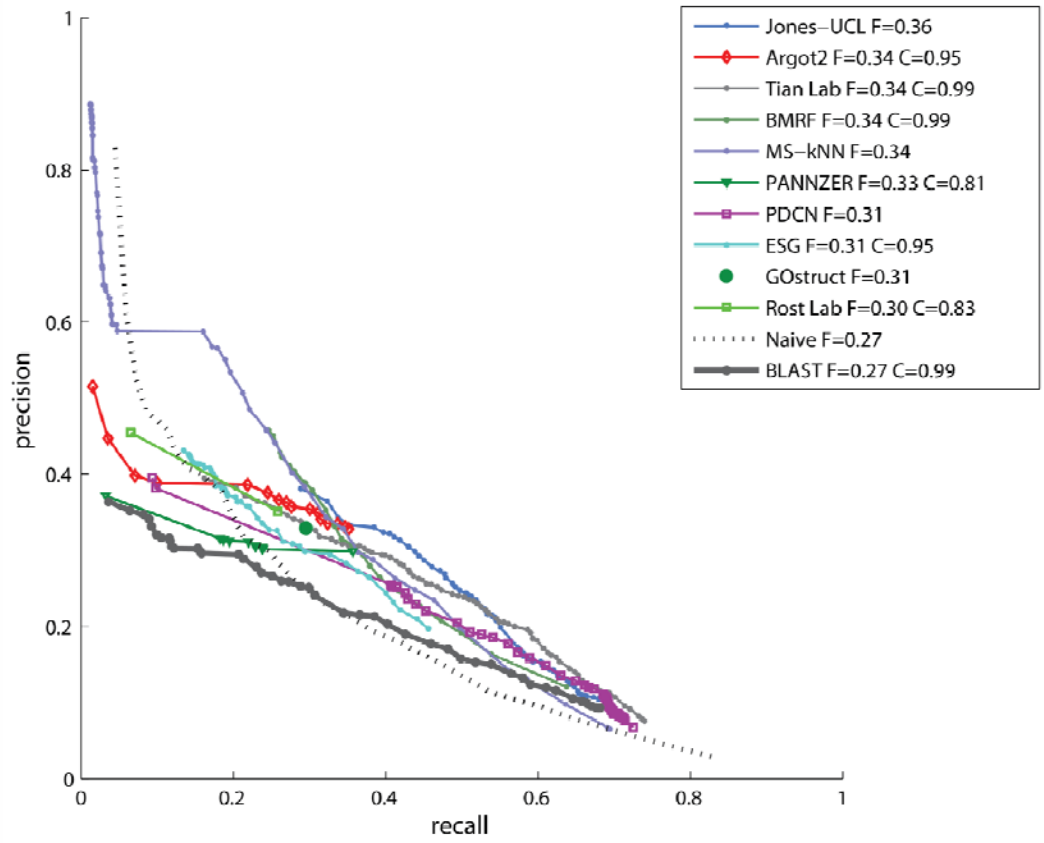
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 6D:



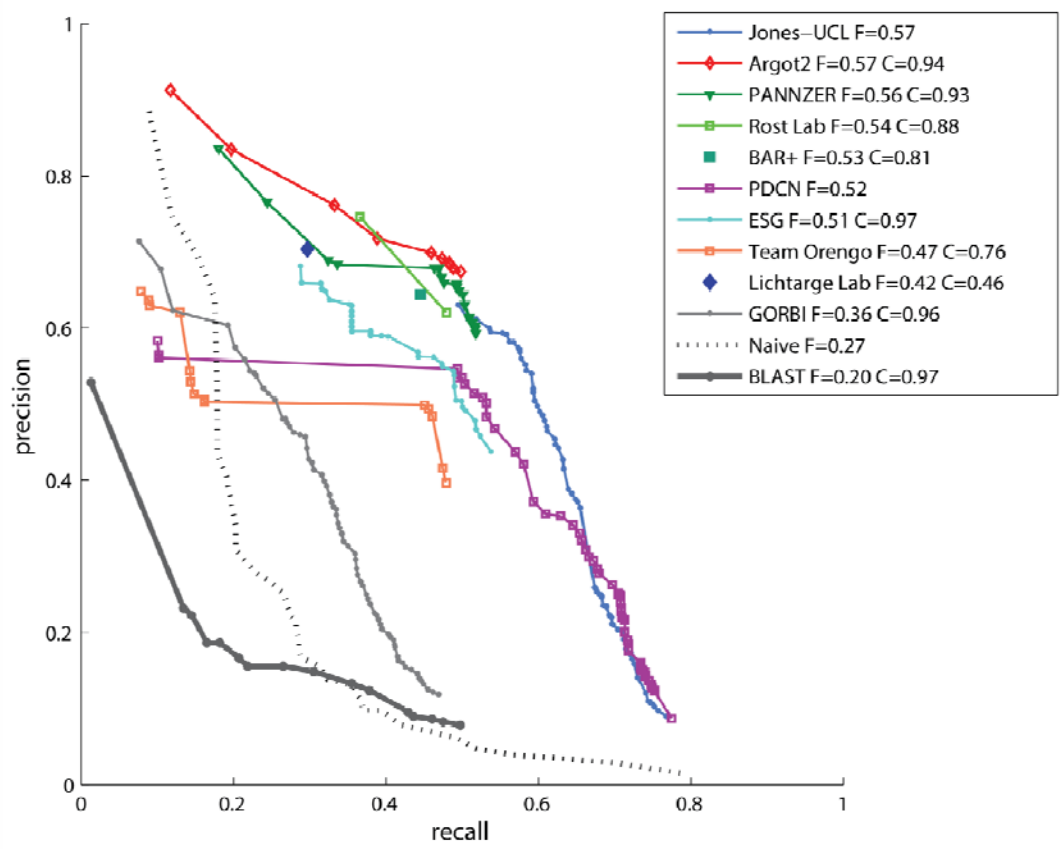
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 6E:



Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 6F:

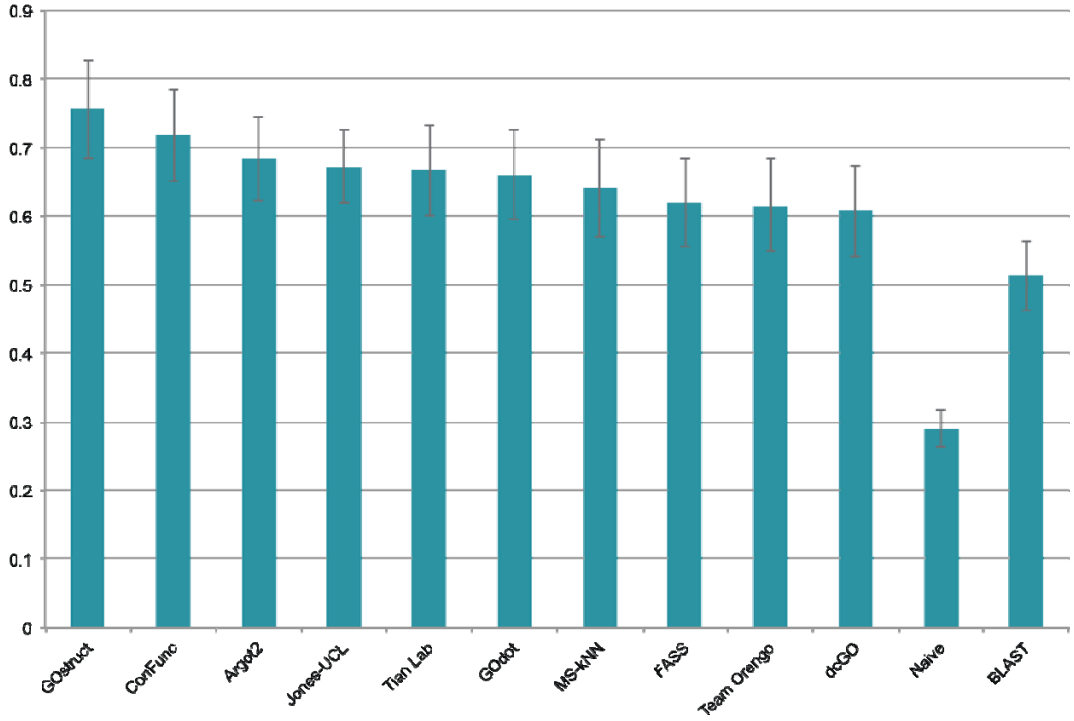


Nature Methods: doi:10.1038/nmeth.2340

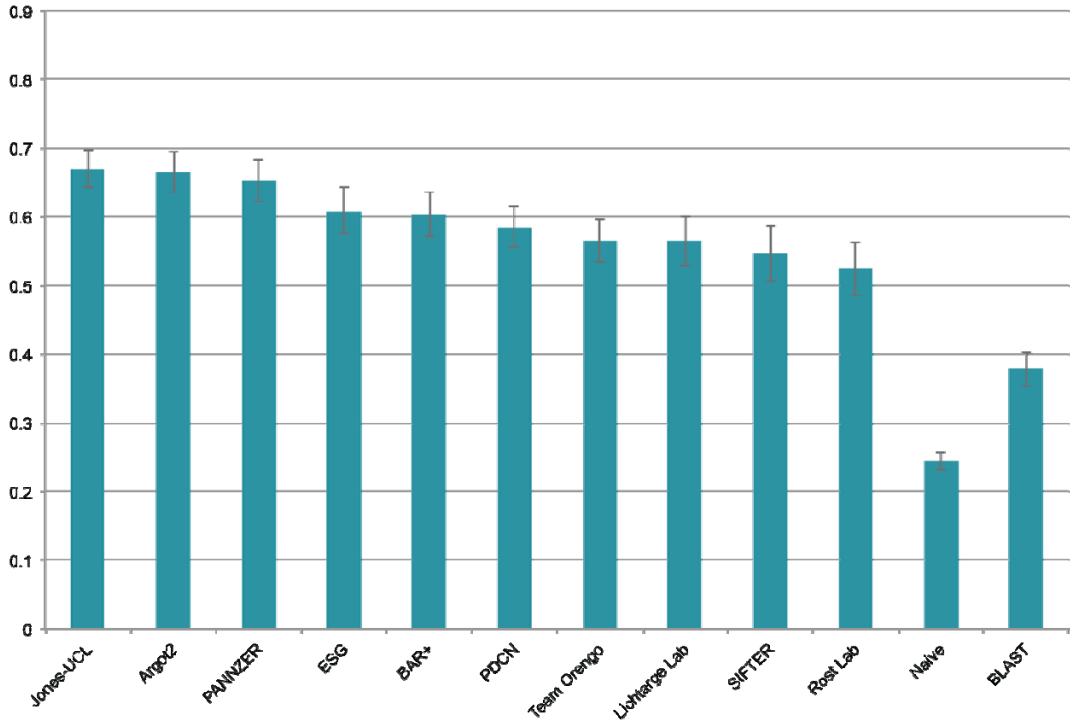
Supplementary Figure 7. Performance evaluation based on the maximum F-measure for the top-performing methods for the Molecular Function ontology (A-E) and Biological Process Ontology (F-J). Targets with Molecular Function annotation "protein binding", as their only leaf term were excluded. Only the species with 30 targets or more are included. All bars show the top ten participating methods as well as the BLAST and Naïve baseline methods. A perfect predictor would be characterized with F_{max} of 1. Confidence intervals (95%) were determined using bootstrapping with 10,000 iterations on the set of target sequences.

Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 7A: Molecular Function – *A. thaliana*.

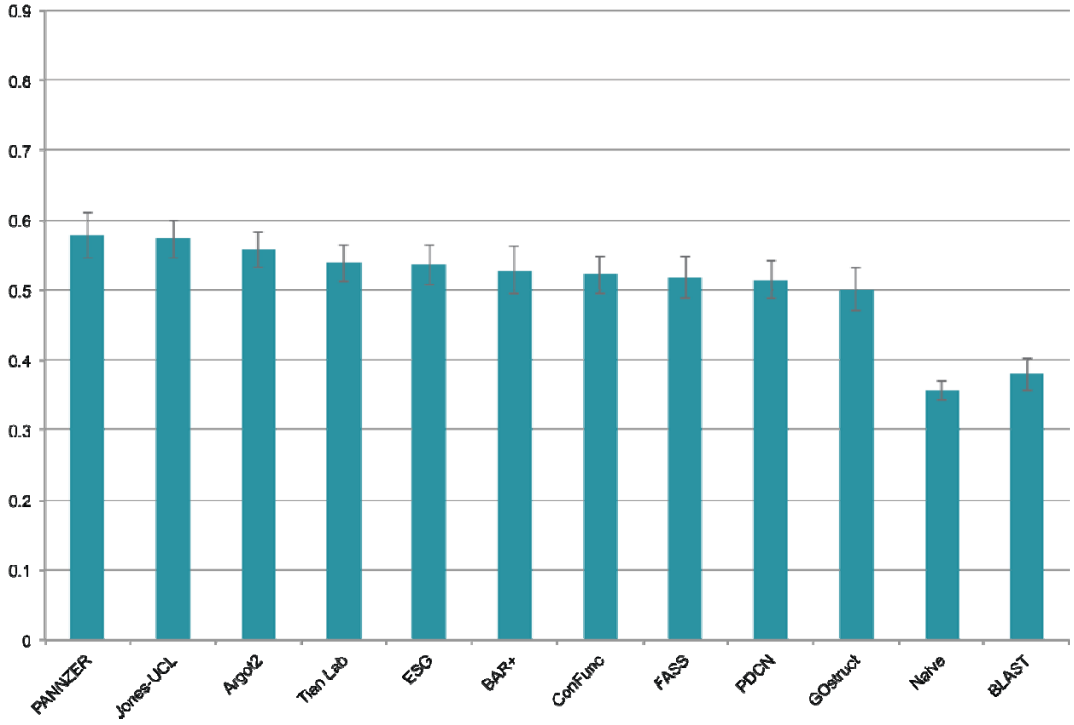


Supplementary Figure 7B: Molecular Function – *E. coli* K-12.

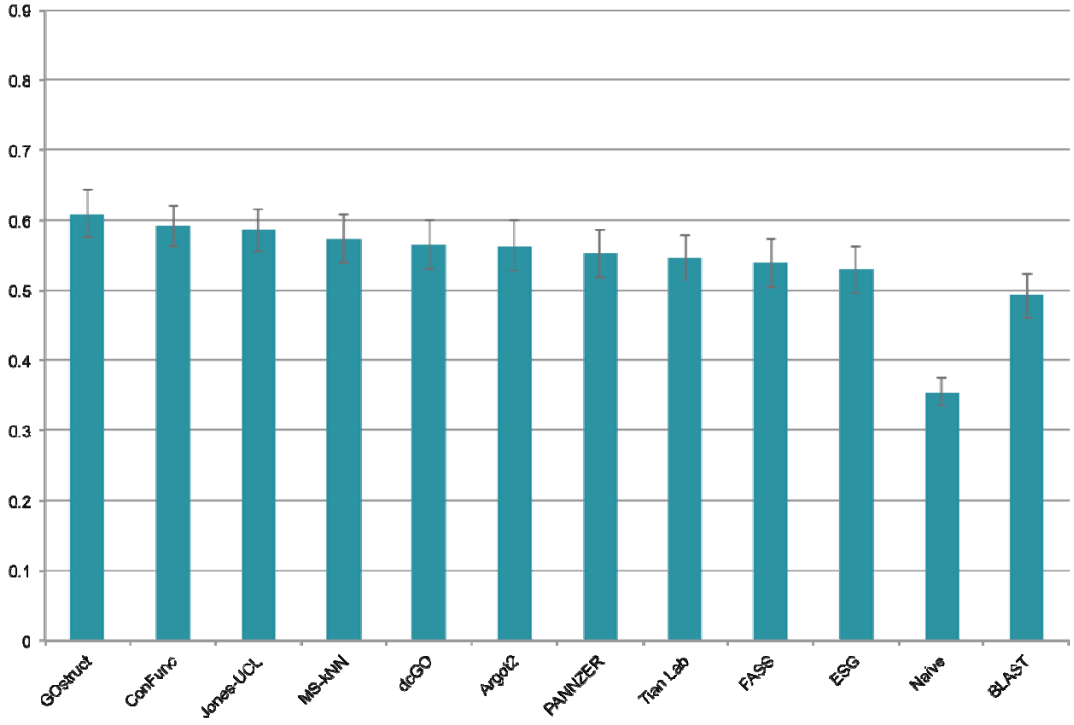


Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 7C: Molecular Function – *H. sapiens*.

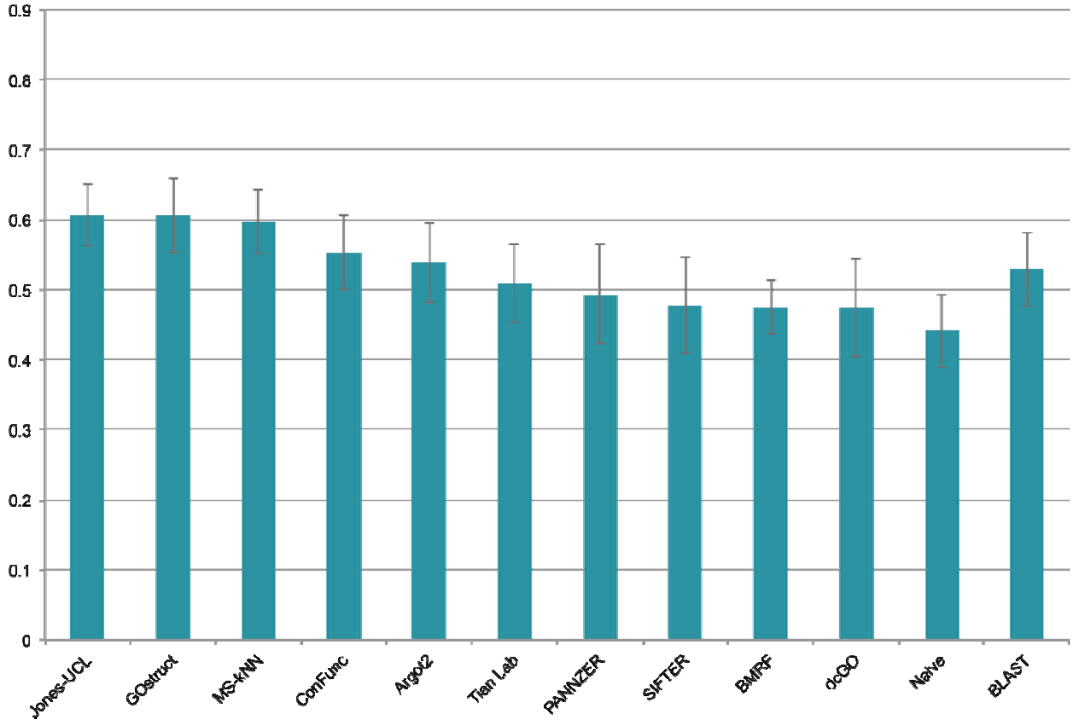


Supplementary Figure 7D: Molecular Function – *M. musculus*.



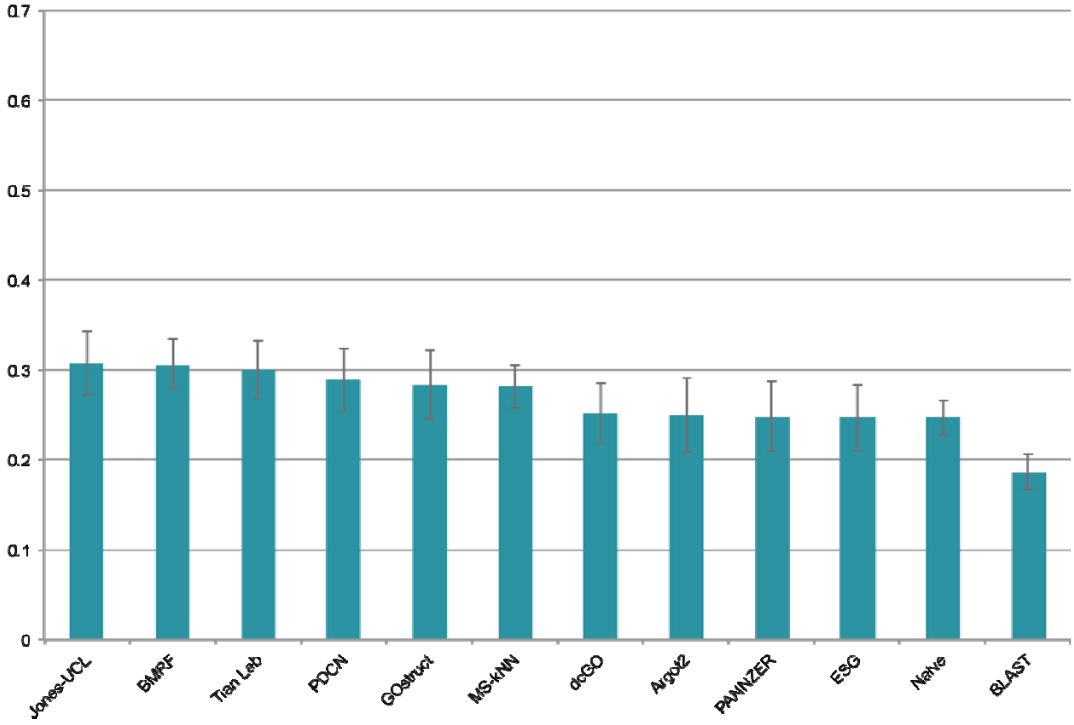
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 7E: Molecular Function – *R. norvegicus*.

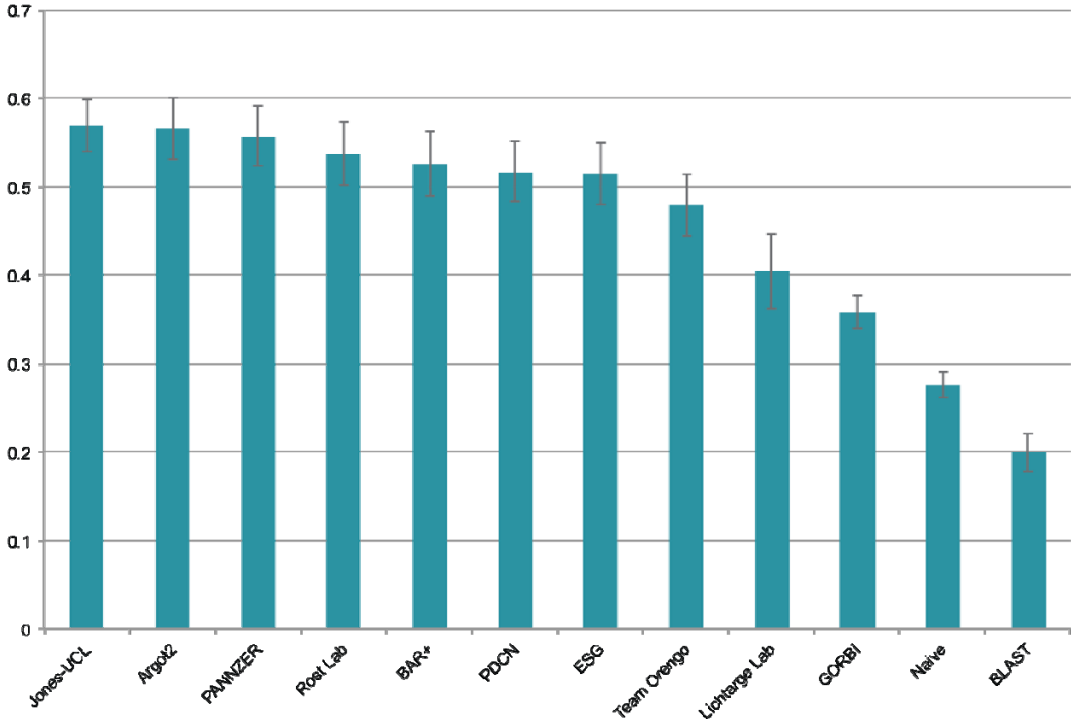


Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 7F: Biological Process – *A. thaliana*.

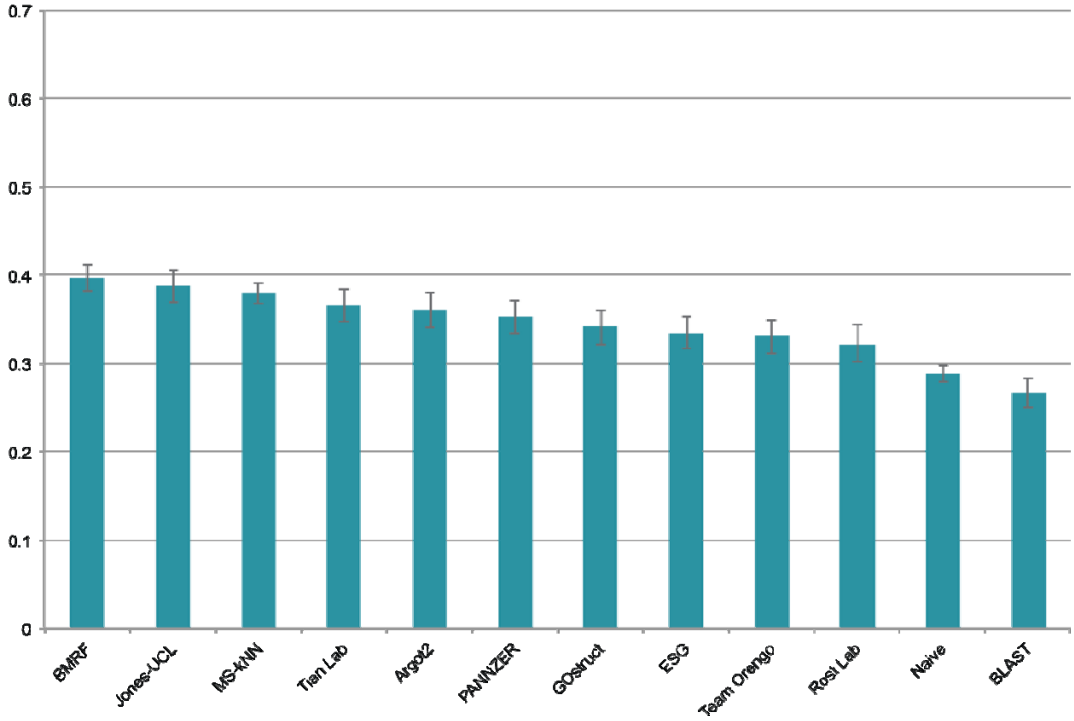


Supplementary Figure 7G: Biological Process – *E. coli* K-12.

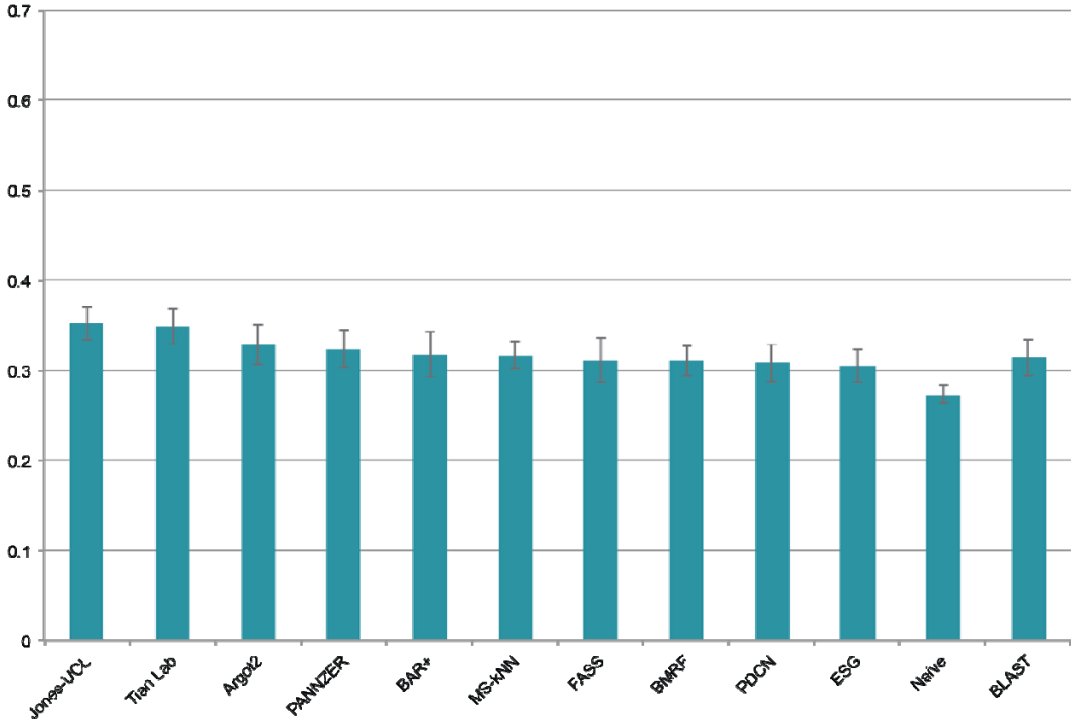


Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 7H: Biological Process – *H. sapiens*.

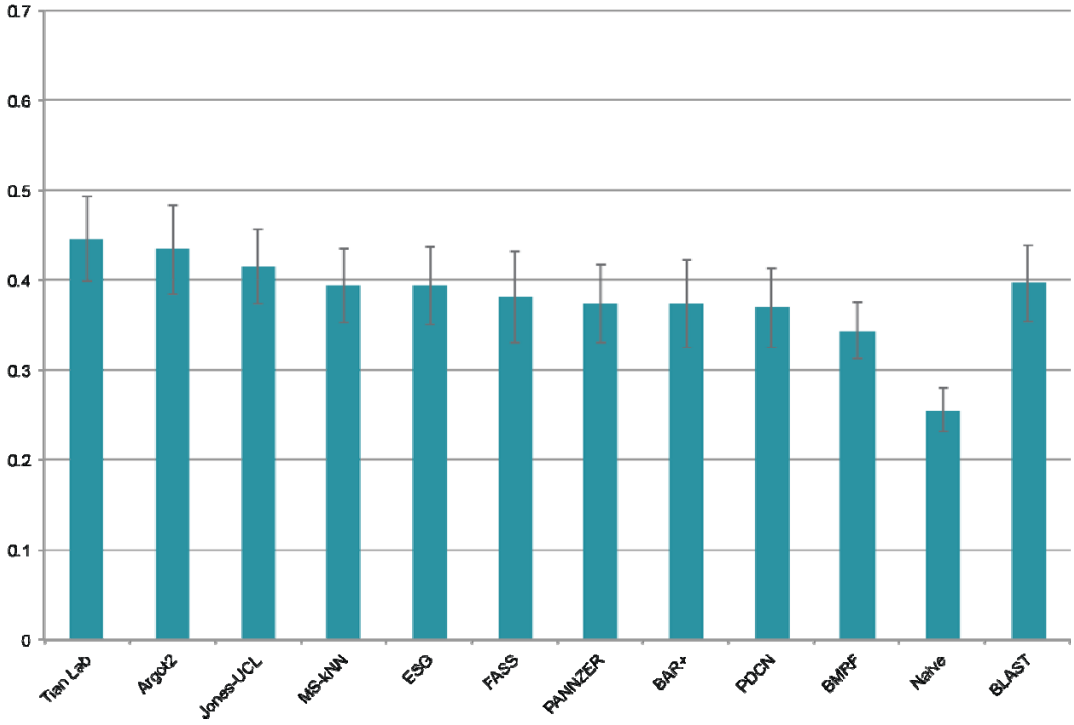


Supplementary Figure 7I: Biological Process – *M. musculus*.



Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 7J: Biological Process – *R. norvegicus*.



Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 8. Weighted precision-recall curves for the top-performing methods for (A) Molecular Function ontology and (B) Biological Process ontology. All panels show the top ten participating methods in each category, as well as the BLAST and Naïve baseline methods. The legend provides the maximum F-measure (*F*) for all methods and coverage (*C*) for all methods that did not make predictions on all targets. In cases where a Principal Investigator (PI) participated with multiple teams, only the results of the best scoring method are presented. Previously unpublished methods are denoted as *experimental*. Methods: MS-kNN[65] (PI: Slobodan Vucetic, Temple University), GOstruct[34] (PI: Asa Ben-Hur, Colorado State University), Jones-UCL[54] (PI: David Jones, University College London), Argot2[55] (PI: Stefano Toppo, University of Padova), PANNZER (PI: Liisa Holm, University of Helsinki; experimental method), BMRF[33] (PI: Cajo J. F. ter Braak, Wageningen University), Team Orengo[60] (PI: Christine Orengo, University College London), ESG[56] (PI: Daisuke Kihara, Purdue University), PCDN[59] (PI: Jianlin Cheng, University of Missouri), BAR+[57, 58] (PI: Rita Casadio, University of Bologna), Rost Lab[66] (PI: Burkhard Rost, Technische Universität München), Tian Lab (PI: Weidong Tian, Fudan University; experimental method).

Calculation of the weighted precision-recall curve. Each term *f* in the ontology was weighted according to the information content of that term. The information content of the term *f* was calculated as

$$i(f) = \log_2 \frac{1}{\Pr(f)}$$

where Pr(*f*) is the relative frequency that a randomly selected protein will be associated with term *f* in the ontology (probabilities were determined based on the Swiss-Prot database). For a given target protein *i* and some threshold *t*, the weighted precision and recall were then calculated as

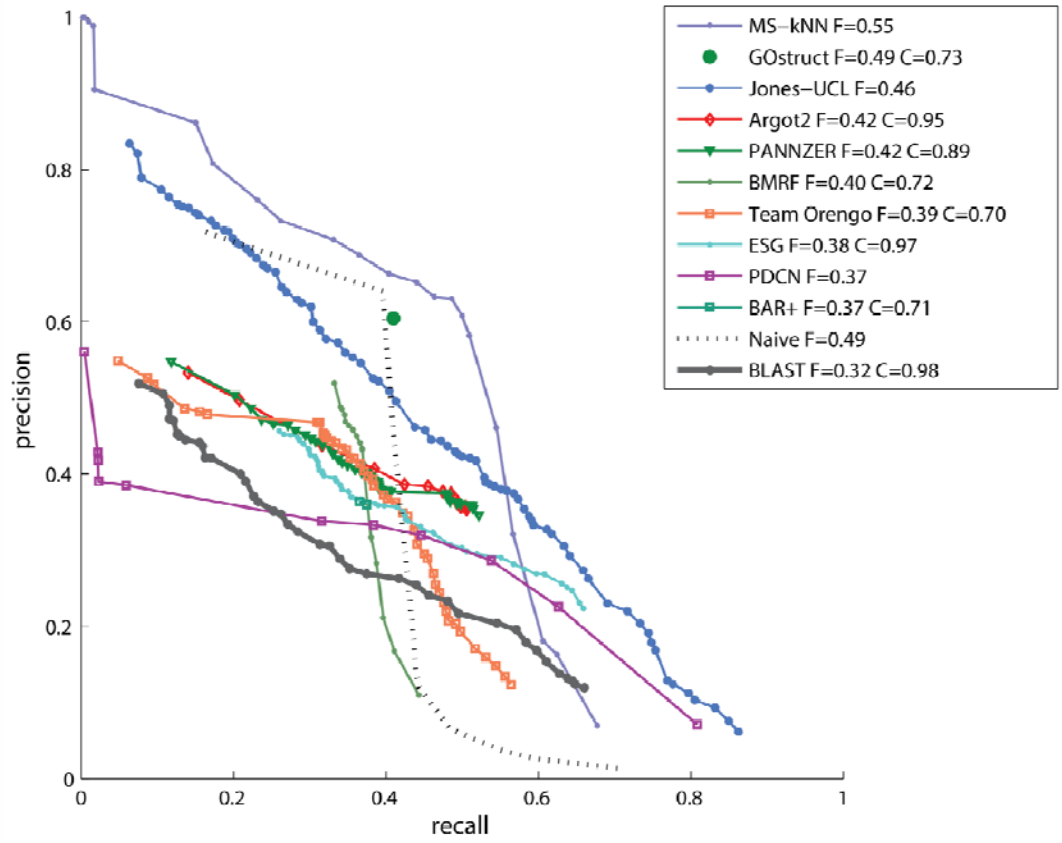
$$wpr_i(t) = \frac{\sum_{f \in T_i \cap P_i(t)} i(f)}{\sum_{f \in P_i(t)} i(f)}, \text{ and}$$

$$wrc_i(t) = \frac{\sum_{f \in T_i \cap P_i(t)} i(f)}{\sum_{f \in T_i} i(f)}$$

where $P_i(t)$ is the set of predicted terms for protein *i* with score greater than threshold *t* and T_i is the set of true terms for protein *i*. For each threshold, the weighted precisions and recalls were then averaged over all proteins and a point in the precision-recall space was created.

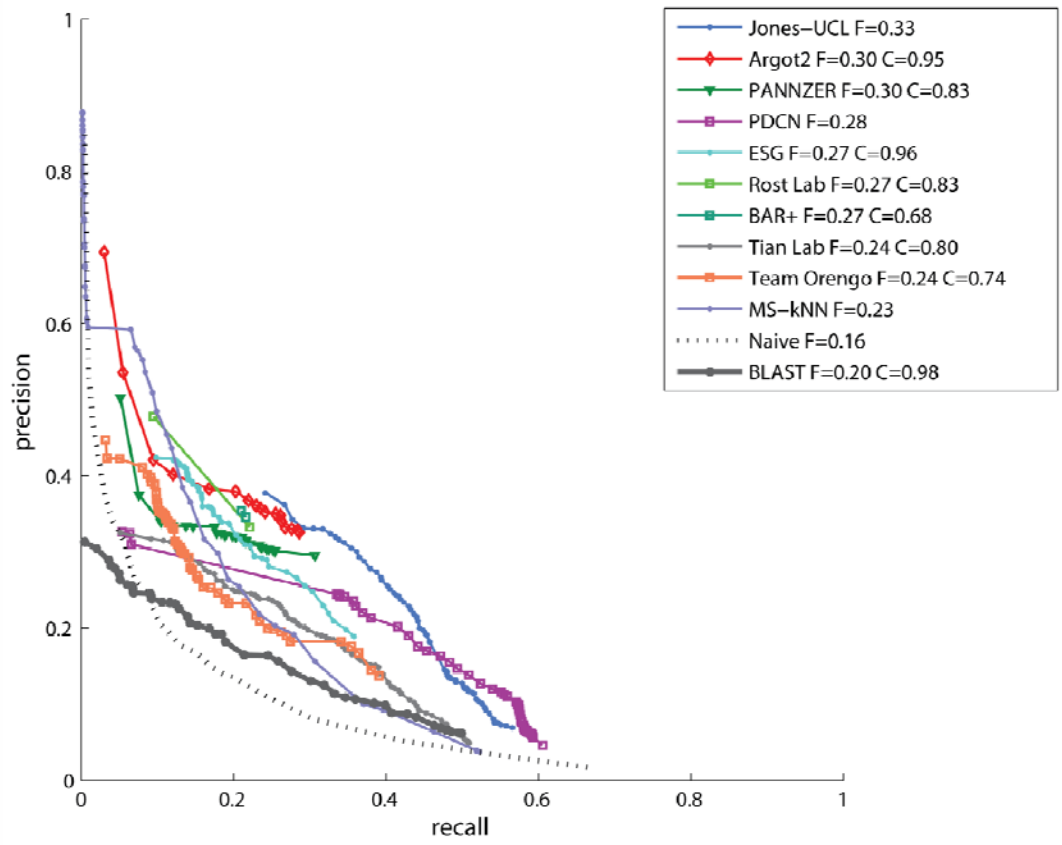
Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 8A:



Nature Methods: doi:10.1038/nmeth.2340

Supplementary Figure 8B:



Nature Methods: doi:10.1038/nmeth.2340

Supplementary Table 3. Participating methods grouped according to Principal Investigators (PIs).*

Principal Investigator	Method name	Model (keywords)	Publications
Cajo ter Braak	BMRF	Model 1 (pi, ge, or, lt)	33, 67
Hagit Shatkay	UDelQueens	Model 1 (lt) Model 2 (lt) Model 3 (lt)	68
Jianlin Cheng	PDCN	Model 1 (ppa, spa, dp) Model 2 (ppa, spa, dp) Model 3 (ppa, spa, dp)	59
David Jones	Jones-UCL	Model 1 (ppa, sp, pi, ge, lt, ml, or) Model 2 (lt, ml) Model 3 (ppa, ml)	54
	FFPred	Model 1 (ppa, sp, pi, ge, lt, ml, or) Model 2 (sp, ml)	69, 70
	FunctionSpace	Model 1 (ppa, sp, pi, ge, lt, ml, or) Model 2 (sp, pi, ge, ml)	
Alberto Paccanaro	Paccanaro Lab	Model 1 (ml, sa, p, pi, lt, or, gi)	
Liisa Holm	Holm Group	Model 1 (sa, pa, or, ofi)	Suppl. Inf.
Steven Brenner	SIFTER	Model 1 (ph, ml)	20, 62
	SINK	Model 1 (ph, ml, ofi)	
Daisuke Kihara	PFP	Model 1 (sa)	15, 71
	ESG	Model 1 (sa)	56
	COCO	Model 1 (sa)	
Julian Gough	Superfamily	Model 1 (spa, pps) Model 2 (spa, pps)	72
	Fang	Model 1 (spa, pps) Model 2 (spa, pps)	
	dcGO	Model 1 (spa, pps) Model 2 (spa, pps)	64
Ingolf Sommer	GOdot	Model 1 (ppa, spa, ml)	73
Stefano Toppo	Argot2	Model 1 (sa, spa)	55, 74
Tomislav Šmuc	GORBI	Model 1 (gc, ml, or, pa)	75
Michael Sternberg	ConFunc	Model 1 (spa)	13
Burkhard Rost	Rost Lab	Model 1 (sa) Model 2 (sa) Model 3 (sa)	66
Christine Orengo	Team Orengo	Model 1 (sa, ppa, spa, ml) Model 2 (sa, ppa, spa, ml)	60
Predrag Radivojac	FANN-GO	Model 1 (sa, ml); not evaluated	16
Rita Casadio	BAR+	Model 1 (sa)	57, 58, Suppl. Mat.
Slobodan Vucetic	MS-kNN	Model 1 (sa, ge, pi) Model 2 (sa, ge, pi) Model 3 (sa)	65
Rajendra Joshi	FASS	Model 1 (or) Model 2 (or)	
Tapio Salakoski	Turku BioNLP	Model 1 (sa, ge, ps, ml) Model 2 (sa, ge, ps, ml, lt) Model 3 (ml, lt)	
Asa Ben-Hur	GOstruct	Model 1 (sa, sp, pi, ml) Model 2 (sa, sp, ml)	34
Olivier Lichtarge	Lichtarge Lab	Model 1 (sa, ml, dp) Model 2 (dp, ps, pps) Model 3 (sa)	61
Weidong Tian	Tian Lab	Model 1 (dp, sa, ppa, spa, sp, ps, gc, cm, ml, or, ofi) Model 2 (dp, sa, ppa, spa, sp, ps, gc, cm, ml, or, ofi) Model 3 (dp, sa, ppa, spa, sp, ps, gc, cm, ml, or, ofi)	Suppl. Inf.

Nature Methods: doi:10.1038/nmeth.2340

*) Each Principal Investigator was allowed to have more than one participating group/team, as long as the teams were largely different. Each team was allowed to have up to three variants of their method; these are referred to as "Models". Prediction teams are associated with publications (if the work was previously published). Methods ranked in the top ten (see manuscript) are also associated with manuscripts or full method descriptions in Supplementary Information.

Keyword table:

Code	Keyword	Code	Keyword
sa	sequence alignments	pi	protein interactions
spa	sequence-profile alignments	ge	gene expression
ppa	profile-profile alignments	gi	genetic interactions
ph	phylogeny	ps	protein structure
or	ortholog	pps	predicted protein structure
pa	paralog	cm	comparative model
sp	sequence properties	ml	machine learning-based method
dp	derived/predicted	ofi	other functional information
		lt	literature

Nature Methods: doi:10.1038/nmeth.2340

Supplementary Note

Methods: Short Descriptions

Note: methods are listed lexicographically based on the PI's last name.

PI: Asa Ben-Hur. The GOstruct method models GO term prediction in the framework of SVMs for structured output spaces, constructing a unified model that takes into account the structure of the GO hierarchy.[34] Our approach is highly flexible, and supports the integration of diverse genomic data, including functional and interaction networks, gene expression data, and is also able to leverage sequence and annotations that are made across species. For the CAFA submission we used a GOstruct model that combines two structured SVMs, one trained using sequence data and GO annotations obtained for a variety of species (sequence was represented as BLAST log E-values and additional features), the other trained using species-specific information that included protein-protein interactions extracted from the STRING database. Additionally, our species-specific classifier for mouse used information extracted from the biomedical literature. Following the CAFA submission we improved this component, and it now provides accuracy comparable to that of a classifier trained on protein-protein interactions. Full details will be provided in the GOstruct manuscript, which is part of the CAFA *BMC Bioinformatics* special issue.[76]

PI: Steven Brenner. Statistical Inference of Function Through Evolutionary Relationships (SIFTER) is a probabilistic graphical model for inferring molecular function of unannotated protein sequences using phylogenomics.[20, 62] Based on phylogenomic principles, SIFTER predicts molecular function for members of a protein family given a reconciled phylogeny and available function annotations. In this model, each molecular function may evolve from any other function, and a protein's function may evolve more rapidly after duplication events than after speciation events. The reconciled phylogeny for a protein family, which discriminates duplication events and speciation events, specifies the tree-structured graphical model used in inference. The major points of the method include the following: (1) given our choice of GO as a source of functional labels, functions are not a simple list of mutually exclusive characters, but are vertices in a directed acyclic graph (DAG); (2) we require a model akin to Jukes-Cantor but appropriate for molecular function; (3) generally only a small subset of the proteins in a family are annotated, and the annotations have different degrees of reliability.

PI: Rita Casadio. BAR, the Bologna Annotation Resource[57] was updated (BAR+[58]; available at http://bar.biocomp.unibo.it/bar2.0). The method relies on the concept that sequences can inherit the same function/s and structure from their counterparts, provided that they fall into a cluster endowed with validated annotations. BAR+ is based on a clustering procedure with the constraint that sequence identity (SI) should be ≥40% on at least 90% of the pairwise alignment overlapping (Coverage, Cov). Depending on the annotation types of the sequences within the cluster, all new targets that fall into a cluster can inherit validated annotations by transfer. For generating BAR+ clusters we analyzed a total of over 13 million protein sequences from 988 genomes and UniProtKB release 2010_05. The BAR+ cluster-building pipeline starts with an all-against-all sequence comparison with BLAST in a GRID environment. The alignment results are

Nature Methods: doi:10.1038/nmeth.2340

then regarded as an undirected graph where nodes are proteins and links are allowed only among chains that are 40% identical over at least 90% of the alignment length. All the connected nodes fall within the same cluster; when a cluster incorporates a UniProtKB entry it inherits its annotations (GO and Pfam terms, PDB structures, SCOP classifications). Within a cluster GO and Pfam terms are statistically validated as previously described;[57] validated terms are those endowed with P-values below 0.01.[57] Clusters can contain distantly related proteins that therefore can be annotated with high confidence and eventually can also inherit a structural template, if present. Structural alignments are provided by a cluster HMM.[58] Following CAFA rules, predictions are carried out only for the Molecular Function (MFO) and Biological Process (BPO) Ontologies. We aligned all the target sequences against BAR+ clusters that, according to our constraints, contain validated GO terms. When a sequence did not match any cluster, singletons (standalone sequences in BAR+) were used to transfer annotation, provided that the criteria were also met. Out of a total of 48,298 sequences, about 63% inherited validated MFO and BPO GO terms (with score 1). Sequences in BAR+ can inherit many other annotation types. In the eukaryotic set 16,428 sequences were also annotated with Cellular Component (CCO) terms. Only for CAFA prediction we tried a coverage constraint ≥70% (SI ≥ 40%) but the number of annotated targets increased only by 3%. These last annotations were, however, submitted with score 0.50 instead of 1.

PI: Jianlin Cheng. PDCN integrated sequence-profile and profile-profile alignment methods (PSI-BLAST and HHSearch) with several protein function databases (Gene Ontology, Swiss-Prot, and Pfam) to predict the functions of target proteins having detectable homology with proteins of known function. For the hard cases where no homologous proteins could be identified, it applied domain co-occurrence networks (DCNs) to make predictions by transferring function annotations between neighboring domains on DCNs.

PI: Julian Gough. Proteins are inherently of modular design, with domains not only as structural and evolutionary units but also as functional units. In spite of the conceptual importance of domains as functional carriers, conventionally we are accustomed to considering whole proteins instead. In multi-domain proteins, two or more domains can be combined together as larger evolutionary units (termed "supradomains"). Functional annotations for supradomains, although essential for genome annotations in higher organisms, remain a challenging obstacle for manual curation that involves looking at the functions of multi-domain proteins they reside in. To meet this challenge, we employ the concept of reverse engineering to make an automatic inference of domain- and supradomain-level annotations from protein-level annotations and protein domain assignments. This approach is specifically tailored to the hierarchical structure of the ontology itself and aims to capture those ontologies/annotations statistically significant for domains (and supradomains). It has been successfully tested on Gene Ontology (GO) for SCOP domain-centric annotations[72] and has been extended here to SCOP supradomain-level GO annotations, to another functional ontology Enzyme Commission, and to species-specific phenotype/anatomy ontologies for human, mouse, worm, fly, zebrafish, yeast and arabidopsis (unpublished). These domain-centric, functional, and phenotypic ontologies and annotations are available at http://supfam.org, which provides a basis for comparative functional and phenotypic genomics involving SCOP domains and supra-domains. In additional to SCOP, a version for InterPro domains is also generated. In all, our method represents a domain-centric

Nature Methods: doi:10.1038/nmeth.2340

solution towards genome annotations, regardless of whether proteins are multi-domain or single-domain, or whether they are of functional or phenotypic relevance.

PI: Liisa Holm. PANNZER methodology (http://ekhidna.biocenter.helsinki.fi/pannzer) starts with a BLAST or HMMer output result (hit list). The list is filtered using thresholds on alignment coverages and sequence similarity. In the CAFA experiment, we applied an additional filter that only retained sequences that have non-IEA GO annotations. The retained hits are re-scored using query and target sequence coverage, sequence identity percentages and the distance in the NCBI taxonomic tree between the query and target species. Scores are weighted using a regression model that was designed to maximize the correlation from sequence search scores to description line text similarities. In the next step, hits are clustered based on the similarities of their description lines. We used Term Frequency-Inverse Document Frequency and cosine similarity for DE similarity. The best cluster is then selected using a Gene Set Z-score (GSZ)[77] which favors clusters enriched at the top of the re-scored hit list. We then predict the significance of each GO class that occurs within the best cluster. We calculate the hypergeometric P-value, GSZ, and an asymptotic P-value for GSZ for each GO class and combine them into a single score using a regression model. This regression model predicts the Jaccard similarity between the candidate GO class and the correct GO class. The score from the regression model is used to sort the reported GO classes. In addition, a significance weight is generated for score values by monitoring the behavior of correct and incorrect GO classes in a training dataset.

PI: David Jones. The Jones-UCL approach combines a wide variety of tools and biological information sources, encompassing sequence, gene expression, and protein-protein interaction data into a single framework. Each component method was separately calibrated to produce precision estimates using a benchmark set of 1546 well-annotated Swiss-Prot entries. For the particular CAFA targets that were evaluated, we found that the high-throughput data-integration-based predictions (FunctionSpace), including analyses of gene expression and protein-protein interaction data, contributed least to the final predictions. The largest contribution to correct predictions came from homology-based function prediction, e.g., from PSI-BLAST sequence similarity, orthologous groups and profile-profile comparisons. Other useful predictions were obtained from a novel Swiss-Prot text mining and trigram residue analysis. Information from all these methods were combined in a probabilistic manner taking into account the GO ontology structure to produce final GO results for molecular function and biological process.

PI: Rajendra Joshi. Functional Annotation using Similarity Search (FASS) employs at the first instance BLASTP[11] search against the UniProt database.[36] Significant matches are filtered using criteria such as %overlap, %identity and E-value. The GO terms of the significant matches thus obtained were assigned to the target dataset. BLAST2GO[78] was also run simultaneously against the nr database in an iterative manner starting from very stringent to relaxed cutoffs. A first run yielded GO terms for a set of sequences with stringent cutoffs. Subsequent reruns were performed for the remaining data with relaxed cutoffs. This ensured optimal coverage of dataset for GO term assignment. Concurrently, E.C. numbers were obtained using the tool EFICAz^2 [79] and the GO terms for the same were retrieved using EC2GO.[80] A consensus approach was then used to infer GO terms from all the three outputs using in-house developed Perl scripts.

PI: Daisuke Kihara. The PFP algorithm[15, 71] uses PSI-BLAST to obtain sequence hits for a target sequence and computes the score to GO term f_a as follows:

Nature Methods: doi:10.1038/nmeth.2340

$$s(f_a) = \sum_{i=1}^{N} \sum_{j=1}^{N_f(i)} (-\log E(i) + b) P(f_a \mid f_j)$$

where N is the number of sequence hits considered in the PSI-BLAST output, $N_f(i)$ is the number of GO annotations for the sequence hit i, $E(i)$ is the PSI-BLAST E-value for the sequence hit i, f_j is the *j*-th annotation of the sequence hit i, and constant b takes value 2 (i.e. $\log_{10} 100$) to keep the score positive when retrieved sequences up to E-value of 100 are used. The conditional probabilities $P(f_a \mid f_j)$ are to consider co-occurrence of GO terms in single sequence annotation, which are computed as the ratio of number of proteins co-annotated with GO terms f_a and f_j as compared with ones annotated only with the term f_j. To take into account the hierarchical structure of the GO, PFP transfers the raw score to the parental terms by computing the proportion of proteins annotated with f_a relative to all proteins that belong to the parental GO term in the database. The score of a GO term computed as the sum of the directly computed score by the equation above and the ones from the parental propagation is called the raw score.

ESG[56] recursively performs PSI-BLAST searches from sequence hits obtained in the initial search from the target sequence, thereby performing multi-level exploration of the sequence similarity space around the target protein. Each sequence hit in a search is assigned a weight that is computed as the proportion of the –log(E-value) of the sequence relative to the sum of –log(E-value) from all the sequence hits considered in the search of the same level, and this weight is assigned for GO terms annotating the sequence hit. The weights for GO terms found in the second level search are computed in the same fashion. Ultimately, the score for a GO term is computed as the total weight from the two levels of the searches. The score for each GO term ranges from 0 to 1.0.

PI: Olivier Lichtarge. In protein structures, we begin by mapping key functional residues by Evolutionary Tracing (ET).[81, 82] ET ranks residues by correlating their sequence variations with phylogenetic divergences. Top-ranked residues cluster in the structure to reveal functional surfaces. To predict function, we then: (a) pick 3D templates of 5 or 6 top-ranked surface ET residues; (b) identify matches in other protein structures in which identical residues have similar functional importance and geometry; and (c) transfer function annotations from known to unknown proteins based on these hits.[83-85] This ET annotation (ETA) approach scales to the structural genome and is specific; it enables substrate-level predictions of enzyme function (at the 4th level of the Enzyme Commission (EC) classification) with 90% accuracy at 66% coverage. At the 3rd EC level accuracy and coverage rise to 96% and 70%. Finally, we note that template matches create a structural proteomic network of ETA hits (the edges) between proteins (the nodes). Having posed function annotation as a graph-based semi-supervised learning problem, it then becomes possible to exploit global rather than local network analysis. Starting with functional labels concentrated initially at nodes that carry these functions, Graph Information Diffusion (GID) redistributes the information in these labels over the entire network to suggest putative function(s) at any node to which a label significantly flows. GID specifically redistributes labels by solving a sparse linear diffusion equation, and its predictions of function were recently experimentally validated.[86]

Nature Methods: doi:10.1038/nmeth.2340

To annotate protein sequences, we sought to generalize GID to networks with millions rather than thousands of nodes so as to include sequence-based information. We introduce a compression technique that eliminates the mostly redundant edges of fully connected subgraphs. This dramatically reduces the number of network links, but not accuracy, allowing GID to analyze otherwise computationally prohibitive amounts of experimental data, such as those in the very large STRING database network,[87] which covers 373 full genomes (version 7.1). GID was more sensitive than other non-local methods over a compressed STRING network and more sensitive than local methods over the full network.

PI: Christine Orengo. The Orengo group used libraries of HMMs that model functional families of protein domains below the CATH[88] superfamily level. These families were produced using a novel pipeline for domain family identification. In brief, this uses the sequence clustering protocol described by Lee et al.[89] in conjunction with high-quality GO annotation data. The workflow for each CAFA target is as follows. The protein is first assigned a set of CATH domains, using the Gene3D[90] superfamily models. At least one domain was found in ~60% of all targets. Each domain is then assigned to one of the families within its superfamily, using the family models. Finally, GO terms are assigned to the target protein as a whole in a probabilistic manner. This is based on the detected domain families, each of which is associated with different GO terms via its seed sequences with varying strength (frequency of sequence-term associations). The term scoring procedure further takes into account the GO DAG structure and the quality of the different domain family assignments (HMM hit scores).

PI: Alberto Paccanaro. Our method combined known GO annotations together with functional information derived from experiments. This was done by diffusing the known GO labels over functional graphs spanning the entire genomes from which the CAFA targets had been selected. First, for each genome, proteins with known GO annotation were assigned their functional labels. Then, functional information derived from experiments (protein-protein interaction, co-expression) was assembled into graphs where each node represented a protein and each link was labeled with the strength of the functional association. These graphs were then averaged into one single functional graph. Finally, the known GO labels were diffused on this graph using the label diffusion method proposed by Zhou et al.[91]

PI: Predrag Radivojac. We apply our model FANN-GO[16] to the target sequences provided for the assessment. FANN-GO stands for Functional ANNotator of GO terms and is an ensemble of multi-output feed-forward artificial neural networks (ANNs). For a given input sequence, we first apply the GOtcha method[14] to generate i-scores for each term in GO. These scores present the input to each neural network. Because of the large number of functional terms in GO, each ensemble member was trained on 100 randomly selected outputs, and the final prediction for a term is provided as an average of scores over all networks that contained an output associated with the term. Two separate models were provided, one for the Molecular Function ontology and the other for the Biological Process ontology. The model was trained on all GO terms associated with at least 50 experimentally annotated proteins in the January 2010 version of the Swiss-Prot database. FANN-GO was not evaluated in CAFA to avoid a potential conflict of interest with the organizers of the experiment.

PI: Burkhard Rost. All three methods exclusively use sequences with existing functional annotations as input. They do so by BLASTing a given target against the GO annotated part of

Nature Methods: doi:10.1038/nmeth.2340

Swiss-Prot and looking at the first *k* hits returned. Methods only differ in the amount and quality of hits they consider and how they assign a probability to each GO term found.

PI: Tapio Salakoski. Our system uses an SVM classifier[92] to predict whether each of the 385 most common GO terms applies for a given protein. We train the classifiers using as features UniProt protein structures and families, precalculated predictions from the Blast2GO tool[78] provided by SIMAP (http://boincsimap.org/boincsimap/), known tissues of expression from UniGene (http://www.ncbi.nlm.nih.gov/unigene) and information on whether the protein belongs to one of the seven CAFA target species. We also evaluate the suitability of text mining for function prediction, by using PubMed-wide text mining events, descriptions of molecular interactions.[93] Our results indicate the SVM classification can improve on the Blast2GO baseline, and that event type text mining provides information usable for function classification but is still unable to outperform other sources of data. Our method constitutes our first work on protein function prediction, and is to be considered experimental. We conclude that while the approach of multiple SVM classifiers is a valid tool for function prediction, more work is required to improve performance, especially the recall of the system.

PI: Hagit Shatkay. We developed a text-based classifier for predicting protein function, using principles similar to those we have used in an earlier work to predict protein subcellular location.[94] To train our classifier, we first compiled a dataset of proteins for which a reliable functional annotation was available from UniProtKB/Swiss-Prot. For each of these proteins, we retrieved the PubMed abstracts referenced from the respective UniProtKB entry. These abstracts formed the source of text features, namely, we extracted from the abstracts characteristic terms for each functional category. Those are terms whose occurrence probability is statistically significantly different in abstracts associated with proteins from one functional class than those associated with all other classes. The set of all characteristic terms was used as text features to represent each protein as a weighted bag-of-words vector. The proteins, represented as text-based vectors, were used to train and test a k-nearest neighbour classifier. We applied the resulting classifier to the proteins from the eukaryotic and prokaryotic track of the CAFA Challenge. CAFA proteins that had PubMed abstracts referenced from their respective UnitProtKB entry were represented as described above. CAFA proteins that had no PubMed references associated with their UniProtKB entries were represented as a weighted combination of the feature vectors of homologous proteins in our dataset.

PI: Tomislav Šmuc. The GORBI method combines the available annotations for groups of inferred cliques of orthologs (OMA groups: http://omabrowser.org/) with the pattern of presence or absence – phylogenetic profiles – of various classes of orthologs and inferred paralogs and presents the data to a decision-tree classifier implemented in the Random Forest (RF) setting (CLUS: http://dtai.cs.kuleuven.be/clus/). We focus on the prokaryotic annotations only, as the phylogenetic profiling method is more powerful on prokaryotic than on eukaryotic data.

PI: Ingolf Sommer. Based on the GOdot method[73] for function prediction from a given protein structure, we transferred the concepts to protein sequences. An input protein sequence is compared to a reference set of proteins. The most similar sequences are identified and their GO molecular function terms are obtained. Local function conservation in sequence space is analyzed for these GO terms. These models return function conservation scores that indicate how

Nature Methods: doi:10.1038/nmeth.2340

conserved the function in the region surrounding the protein is and denote the confidence in the predicted term.

Here, the UniRef90, a compressed version of the UniProtKB, was used as the reference protein set. Profile Hidden Markov Models are employed to compare protein sequences. The iterative search jackhmmer from the HMMer 3 package[95] was used for the detection of homologs in the reference set.

For a query sequence, GO terms from the ten most similar sequences are considered. For each GO term the sequence distances to the nearest 200 proteins according to jackhmmer are related to known annotations of that GO term. Logistic regression is used to model the local function conservation in the sequence neighborhood, following the approach in GOdot. The logistic regression models are then used to compute function conservation scores for the individual GO terms.

PI: Michael Sternberg. Predictions by ConFunc were based on the ConFunc method.[13] ConFunc is a protein-sequence-based method, which identifies homologues of a query sequence and groups them into subalignments according to their GO annotations. Predictions are made by comparing the query sequence to conserved residues present in each of the subalignments. CAFA predictions were supplemented by using other computationally derived annotations based primarily on mappings from other annotation sources (UniProt-GOA).

PI: Cajo ter Braak. We developed a probabilistic method for protein function prediction that is based on Bayesian Markov Random Field (BMRF) analysis. BMRF propagates the annotations of proteins through one or multiple networks of protein associations, using an adaptive Markov Chain Monte Carlo algorithm. Our method is able to integrate diverse types of data such as sequences and networks and in previous studies we have shown that integration of diverse information can lead to improvement of prediction performance.[33] For CAFA, we predicted functions for the proteins coming from the eukaryotes by integrating protein interaction and orthology information. Regarding orthology, we first constructed profiles of memberships in clusters of orthologous groups for all the proteins of the target species, using information from ProGMap.[96] We then used penalized regression to build classifiers for GO terms, using the annotated proteins as a training set and their membership profiles as predictors. GO term predictions for the target proteins were obtained using those classifiers and further integrated with protein interaction information. In particular we constructed a network of protein interactions using data from STRING. In our BMRF model the total probability of a target protein to perform a particular function (GO term) depends on the functions of the direct neighbors of the protein according to the network topology and also on the predictions based on the orthology information. Our workflow is described in further detail in the original publications.[33, 67]

PI: Weidong Tian. In CAFA we employ a combined approach for protein function prediction (CAPFP). There are two major component algorithms in CAPFP: one that predicts protein function solely based on protein sequence (sequence-based), and the other that predicts protein function by integrating omics features associated with that protein (feature-based). In the sequence-based algorithm, we first apply PSI-BLAST to search for homologous sequences associated with a given protein sequence. Then, we prepare a multiple-sequence alignment

Nature Methods: doi:10.1038/nmeth.2340

(MSA) based on PSI-BLAST output, from which we identify functionally discriminating residues (FDRs) specific to a given function. Finally, we predict the function of the protein based on the absence or presence of those FDRs. In feature-based algorithms, we first collect genomic features associated with the proteins for each organism, including protein domains, TFs, orthologs, etc. Then, we use the random forest (RF) algorithm to integrate genomic features and build a prediction model for each GO term. Finally, we predict the functions of a given protein using the RF models. Both the sequence-based and the feature-based algorithms are trained independently. The prediction scores from both models are normalized to 0-1, with higher score indicating higher confidence. For a given function, the final prediction score of a given protein is the higher score from the two models.

PI: Stefano Toppo. Efficient functional annotation of entire genomes is an important step to understand the gene product functions and to provide support for the interpretation of experimental results. The Gene Ontology (GO) has provided the means to standardize annotation classification with a structured vocabulary, which can be easily exploited by computational methods. Argot2 is a web-based distributed function prediction tool able to annotate nucleic or protein sequences on a genomic scale. It accepts as input a list of sequences in FASTA format that are enriched using the results of BLAST and HMMer searches. These sequences are then annotated with GO terms retrieved from the UniProtKB-GOA database and terms are then weighted using the e-values from BLAST and HMMer. The weighted GO terms, which can also be provided directly, are processed according to both their semantic similarity relations described by the Gene Ontology and their associated score. The algorithm has been employed and heavily tested already during in-house genome projects of grape and apple and has proven to have a high precision and recall in our benchmark conditions. The server is freely accessible at http://www.medcomp.medicina.unipd.it/Argot2.

PI: Slobodan Vucetic. We used a weighted variant of the k-nearest neighbor (k-NN) algorithm to calculate a likelihood that protein *p* has function *f*. The prediction score of a protein *p* having function *f* is calculated as

$$score(p, f) = \sum_{p' \in N_k(p)} sim(p, p') I(f \in \text{functions}(p'))$$

where $sim(p, p')$ denotes the similarity score between proteins *p* and p', $I(\cdot)$ is an indicator function that returns 1 if p' is experimentally annotated with *f* and 0 otherwise, and $N_k(p)$ is the set of k nearest neighbors of *p* according to metric $sim(\cdot)$. The similarity scores between two proteins for different data sources were calculated in the following way. For a protein sequence data source, the similarity score was calculated as percent identity divided by 100. For a microarray data source, we used Pearson correlation between normalized gene expressions to measure the similarity score between two proteins. In protein-protein interaction (PPI) data source, the similarity score was set to 1 if the two proteins interacted and 0 otherwise. For each pair (*p*, *f*), we obtained several scores: in particular, one score for sequence data, $score^{SEQ}(p, f)$, and one using PPI, $score^{PPI}(p, f)$. For *J* microarray datasets, we obtained *J* gene expression scores, $score_j^{EXP}(p, f)$, $j = 1, \ldots, J$. We considered several approaches to integrate these $J + 2$ scores for a pair (*p*, *f*) by likelihood maximization and large margin approaches. Interestingly, in our empirical experimental results, none of these approaches worked consistently and

Nature Methods: doi:10.1038/nmeth.2340

significantly better than the simple averaging integration. So, in the CAFA challenge, we use the average integration approach shown below

$$score(p, f) = \frac{1}{3} \cdot score^{SEQ}(p, f) + \frac{1}{3} \cdot score^{PPI}(p, f) + \frac{1}{3J} \cdot \sum_{j=1}^{J} score_j^{EXP}(p, f).$$

Nature Methods: doi:10.1038/nmeth.2340

Methods: Detailed Descriptions

Note: methods are listed lexicographically based on the PI's last name.

Holm Group: PANNZER method in CAFA challenge

Background

Reasons for using descriptions in PANNZER in CAFA challenge

PANNZER is a sequence analysis package that was originally designed for the prediction of gene names or free-text descriptions. The method was modified for the CAFA competition so that it generates Gene Ontology (GO) predictions. The outline of the original PANNZER method is described in Sections A and B with only slight modifications made for CAFA competition. Section C describes the added part that was used to generate GO predictions.

Motivation

PANNZER is a weighted K-nearest neighbor classifier, which uses BLAST to search a protein sequence database and generate a list of neighbors of the query sequence. The central task in functional annotation is the selection of the relevant annotation features from all annotation features that occur in the sequence search list. Annotation features include features such as free-text functional descriptions, gene names, GO classes, Enzyme Commission (EC) classifications etc., but here we focus on GO classes for simplicity. The sequence search list analysis task is similar to the K-nearest neighbor type analysis from data mining, where we have a multidimensional ball with radius R that is drawn around the query sequence (Fig. S1). Next, all the sequences, that are found within the ball (sequences that are closer to query than R), will be further analyzed. Problems for such analysis are caused by: (i) variations in the GO class sizes; (ii) variations in the number of sequences found in sequence search list; (iii) variations in the strength of sequence similarities between the query sequence and the sequences accepted to the analysis; and (iv) occasional misannotated hits with strong similarity contained in the result list. Furthermore, the weighting scheme used should take into account: (v) the number of GO class members in the sequence search result list, and (vi) the sequence similarity of GO class members to the query sequence. Typical sequence query annotation is based on the best matching hit, which is very sensitive to problem (iv) above. If problem (i) is not taken into consideration, then the prediction will automatically report the largest, but not necessarily closest GO classes, an artifact often seen in functional annotation. If problem (ii) is not taken into consideration, then the sequences with larger sequence search lists will have higher risk of false positive GO annotations. If problem (iii) is not taken into account, then the sequences at the end of the sequence search list will have the same impact as those near the top of list. This can again generate false positive GO annotations. Previous solutions based on the hypergeometric P-value, the Jaccard similarity coefficient, the sum of sequence similarity scores for GO class members, or the mean of sequence similarity scores for GO class members have addressed some but not all of the above problems.

Nature Methods: doi:10.1038/nmeth.2340

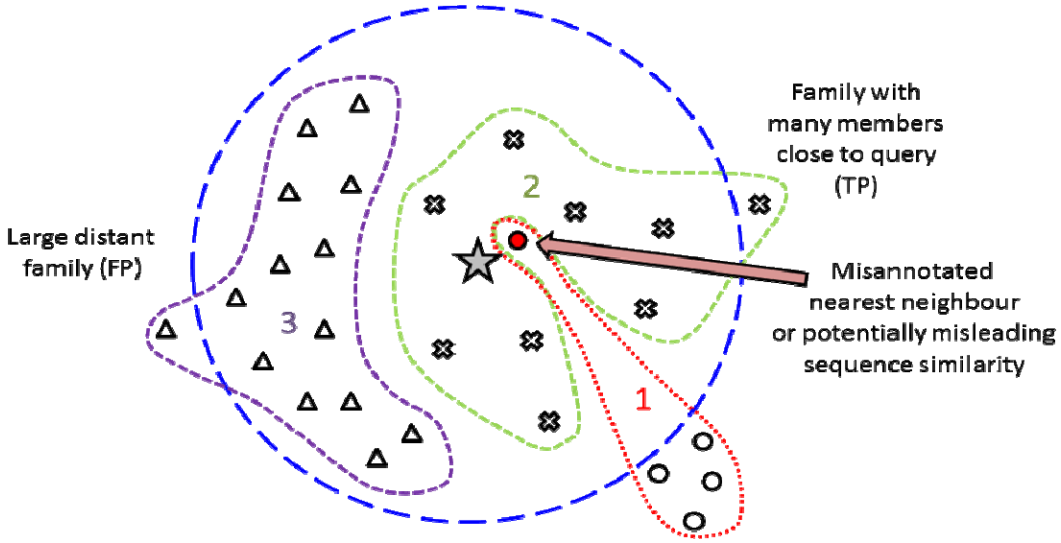


Figure S1. Toy example of similarity search results with query sequence (star). Family 1 (surrounded by red dotted line) has small ratio of family members in similarity search results (blue line). Family 2 (green dotted line) and Family 3 (purple dotted line) have all the family members in search results. Small fraction of found results in Family 1 could indicate that members in results could in reality be members of different families. Query in this case adopts annotation from Family 2, because Family 1 annotations in results are uncertain and because Family 2 is highly represented in query neighbourhood by distance and number.

We propose the PANNZER method that solves all of the problems described (i-vi). PANNZER is a weighted K-nearest approach where the weights are given by the Gene Set Z-score (GSZ).[77] Fig. S1 depicts a case where a simple K-nearest method would pick Family 3 since it is the largest family in the sample. However, it dominates the result list only by size. Family 2 is closer to the query than Family 3, but is smaller in size. When using weighted K-nearest method (using distance as weight), Family 2 will be chosen with the appropriate weighting parameters. A statistical method that also takes account of the background (e.g. hypergeometric P-value, Fisher's exact test, Binomial distribution, Jaccard coefficient of similarity, GSZ) will be needed in order to avoid Family 1 kind of cases to be adopted as a predicted annotation. To our knowledge, GSZ is the only score for weighted K-nearest neighbor analysis that is founded in rigorous probability calculus rather than heuristics.

The PANNZER method

An overall outline of the PANNZER analysis pipeline is shown in Fig. S2. The pipeline has three sections: A, B and C. Section A focuses on the selection of sequences that are relevant for the annotation of query sequence. Section B creates clusters from the selected sequences using similarity of the description lines of the sequences and chooses the best cluster for annotation. Section C collects the GO classes from the chosen cluster and gives them weight scores to

Nature Methods: doi:10.1038/nmeth.2340

estimate the probability that gene has the GO class in question. These steps are described more in detail below.

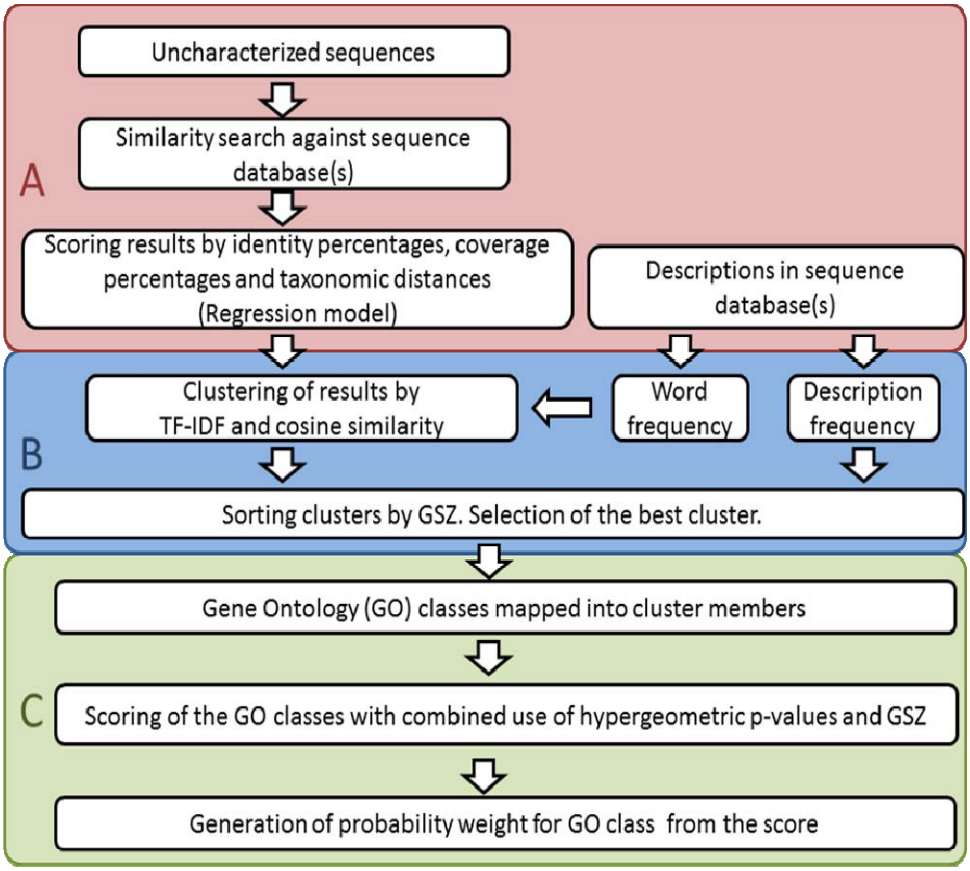


Figure S2. PANNZER dataflow. The unique features of the PANNZER pipeline are weighted K-NN analysis, re-scoring of sequence matches by using query and target coverage, usage of taxonomic distance to weight the sequence matches and usage of descriptions to cluster the selected sequences.

Section A: Sequence level analysis

Sequence search

The first step of a PANNZER analysis is a similarity search against sequence databases. The obtained BLAST result lists can vary a lot in size, from a few to several hundreds of hits. We observed that a very large size of result list was sometimes misdirected to a large neighbouring sequence cluster. Therefore, we limit the number of sequences taken to the analysis and focus only on the sequences that obtained the strongest results from the sequence scoring. The number of sequences selected for analysis was limited to 50. This counting monitored only the sequences with informative descriptions and sequences and with no skipped GO classes. Note that the uninformative sequences are still included in the later analysis steps. The selection of informative

Nature Methods: doi:10.1038/nmeth.2340

descriptions is based on Information Density Score (IDS) with an empirically selected threshold value of IDS = 10:

$$IDS = \frac{1}{n}\sum_{i=1}^{n} idf(t_i)^2$$

where $idf(t)$ is an Inverse Document Frequency score for a term t in description and n is the length of a description d:

$$idf(t, D) = \frac{|D|}{1 + |\{d \in D : t \in d\}|}$$

where $|D|$ is the total number of descriptions in the corpus (i.e. database), and $|\{d \in D : t \in d\}|$ is the number of descriptions (d) where the term t occurs. In order to avoid illegal division by zero, we add one to the equation.

Re-scoring sequence hits

Every remaining result from the sequence similarity search is re-scored using query coverage of alignment in query sequence, coverage of alignment in target sequence and identity percentage from the aligning area. Re-scoring was done since values given by the BLAST (E-value, bit score) gave too significant scores for cases with little significance for annotation transfer. In addition, we included a weight score obtained from the taxonomic distance (see: Non-linear scoring for Taxonomic Distance). All these values are weighted by a regression model and then summed into a single score.

The weights for the combination of the scores were obtained with sparse regression. Besides the original scores, we also included variables multiplied with each other for the combined effects. Furthermore, we included each variable with three different pre-processings: (a) raw variable, (b) square root, and (c) log of the variable. These modifications also allow the modelling of the non-linear behaviour in the regression. Regression was done against the Description Distance Measure (DDM) (see: Clustering of results using TF-IDF and Cosine Similarity) using LASSO and Leaps packages for R language. Here, Leaps performed better. Leaps searches the optimal sparse regression model, where only K variables can obtain non-zero weights. K was allowed to vary from 1 to 10 and an optimal K was selected by monitoring the average performance across 10 replications of seven-fold cross-validation (CV). The quality of the models was assessed with the correlation between the predicted and correct DDM distance and by looking the stability of the model parameters across CV runs (details omitted).

Non-linear scoring for Taxonomic Distance

During the development of PANNZER, we observed that sometimes a larger group of sequences from evolutionally far-away species was informative, for the purposes of function transfer, than a smaller group of sequences from evolutionarily closer species. Therefore, we decided to include

Nature Methods: doi:10.1038/nmeth.2340

inter-species evolutionary distance to our analysis. Unfortunately such measures have weak coverage over species and we decided to use the distance in the NCBI taxonomic tree as an approximation of evolutionary distance. Distance in the NCBI taxonomy tree was calculated by backtracking from query and target species leaf nodes towards the root node until a common ancestor found such that each edge adds one to the distance.

Raw taxonomic distance had very weak correlation with DDM and we speculated that this is caused by non-linear correlation between two variables. This was later confirmed by scatter plots of two variables (data not shown). Therefore we decided to create a non-linear scoring for taxonomic distance. We used the sparse regression (explained below) to learn the weights (*a*, *b*, *c*, *d*) of the equation:

$$Score = a \times Dist + b \times Dist^2 + c \times Dist^3 + d \times \log(Dist + 1)$$

where *Dist* refers to taxonomic distance in NCBI taxonomic tree.

Learning was done against DDM. As a result, we obtained a profile of weights for every taxonomic distance. The obtained profile behaved in the desired fashion by giving a weak weight to taxonomic distance 0 (same species, paralogs), then rising up for nearest species and finally dropping close to minimum at further distance (around distance 30). Unfortunately the weight went up for very large taxonomic distances. This was corrected by fixing the weight to be constant minimum for all taxonomic distances larger than 35. Figure S3 shows the final weight profile. This weight was used in regression to emphasize the sequence matches at the close taxonomic distance.

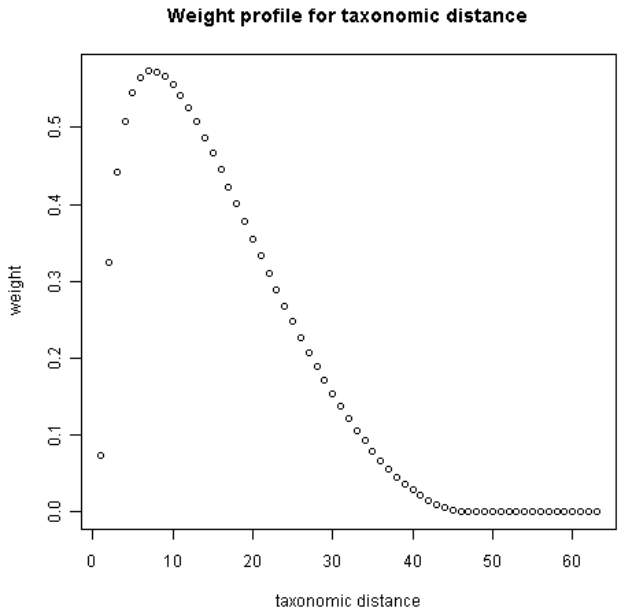


Figure S3. The truncated weight profile for taxonomic distance. Profile was obtained using sparse regression. The only manual correction was fixing of the weight values to minimum for large taxonomic distances.

Nature Methods: doi:10.1038/nmeth.2340

Section B: Clustering using descriptions

Clustering of results using TF-IDF and Cosine Similarity

Term Frequency – Inverse Document Frequency (TF-IDF) is one of the most widely used weighting schemes in information retrieval and text mining. In TF-IDF, the importance of a word increases in relation to the number of times a word appears in the document but also inversely to the frequency of the word in the corpus, i.e., count of descriptions where word appears in database. The rarer the word is in the corpus, the higher is the information value. This weight is a statistical measure used to evaluate how important a word is to a document in a database or corpus:

$$tfidf(t,d,D) = tf(t,d) \times idf(t,D)$$

where $tf(t,d)$ is the frequency of a term *t* in document *d*.

Descriptions are turned into TF-IDF vectors where each element in the vector corresponds to a TF-IDF score of a term. The similarity of two descriptions is calculated by using cosine similarity measure with TF-IDF vectors. Cosine similarity is a dot product of TF-IDF scores of common terms between two descriptions. Cosine similarity is calculated with following formula:

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} A_i^2 \times \sum_{i=1}^{n} B_i^2}}$$

where *A* and *B* are given vectors. Cosine similarity in this article is called Description Distance Measure (DDM).

Sequences with similar descriptions (with empirical DDM threshold 0.8) were clustered together with hierarchical clustering using average linkage.

Selection of best cluster with GSZ

The next task in the PANNZER pipeline is to select one of the created description clusters as a best representative for the query sequence. This is the central task in the sequence feature annotation that occurs while selecting relevant descriptions, GO classes, cellular localizations etc. from sequence search result list (SSRL). We define a relevance score that is used to separate good representative clusters from the randomly occurring clusters. Inputs from the earlier steps to this analysis step are: Size of the description cluster, size of the SSRL, number of sequences that have one of the descriptions that formed a description cluster in the whole database and sum of sequence similarity score (SSS) values for sequences that formed the cluster.

The selection of the best cluster is based on the use of the Gene Set Z-score (GSZ). It was earlier used to analyze GO classes in Gene Set Enrichment Analysis, where it outperformed many competing methods. GSZ is an equation that calculates the sum of the sequence similarity scores for description cluster members and next normalizes the sum score using mean and standard deviation (STD) estimates obtained for the sum under assumption that all the descriptions are

Nature Methods: doi:10.1038/nmeth.2340

randomly distributed A) between SSRL and whole database and B) also within the SSRL. This is our Null Model that defines the situation where there is no signal (description cluster has no connection to the gene in question). Analytically the mean and STD could be obtained with following steps:

1. Repeat the following permutation steps, for example, 1000 times:
    - Distribute all the sequences with the same description as in description cluster randomly between the SSRL and the remaining database. This gives us a random size (K) for description cluster.
    - Select K cluster members randomly from the SSRL list
    - Calculate the sum of SSS values for the selected cluster members
    - Store the obtained sum into permutation result vector
2. Calculate the average value and standard deviation from the permutation result vector

Asymptotic equations, used in GSZ, represent a shortcut that removes the need of CPU consuming permutation runs. The permutation run was presented here to clarify the null model.

GSZ has many useful features: It takes into consideration the frequency of the descriptions in the database and the size of SSRL similarly to the hypergeometric P-value. However, GSZ differs from the hypergeometric P-value by taking the actual SSS values into account. It has also performed well in comparisons with other methods in this context (unpublished research). PANNZER selects the description cluster with highest GSZ score to further analysis

Section C: GO data analysis

GO term mapping to cluster members

GO terms with evidence codes, other than IEA (Inferred from Electronic Annotation), where associated with all the proteins in the GOA database. This gives us the information on both the frequency of GO class in the selected best description cluster and the frequency of GO class in the whole database. Aim is to find GO classes that are detailed enough to show useful biological information but not too detailed as that increases the risk of false positive annotations. Furthermore, this should depend on the information content of the analyzed data.

Scoring of GO terms with a combination of statistical scores

Next, we generated a scoring for the GO classes that occur in the best description cluster. Here we again aim to separate the randomly occurring and frequent clusters from truly informative GO classes. The analysis task was already discussed in chapters Motivation and Selection of best cluster with GSZ. Again the analysis challenges are same.

There is an added complexity level with GO classes, caused by sparseness of curated GO data. The sparseness causes varying numbers of genes without any GO annotation. These genes can be

Nature Methods: doi:10.1038/nmeth.2340

considered as unclear cases and they can be omitted from the analysis. Or they can be included to the pool of negative cases (GO class non-members) in the result list. The latter option gives weaker results as the number of unknown sequences grows in the SSRL, whereas the first option gives exactly the same result.

We decided not to select a single score for GO class scoring, as there are several alternative scores, like the hypergeometric P-values and GSZ for the scoring of GO classes, and each of the scores can be calculated either by including un-annotated genes to the analysis or not. Instead, we used again a sparse regression method to find an optimal weighted combination of different statistical scores.

The following statistical scores were used as an input in this regression: GSZ without prior in the variance, GSZ with a prior in the variance (see Toronen et al.[77] for details on prior), log10(P-value) from hypergeometric distribution for over-representation (upper tail P-value), log10(P-value) from hypergeometric distribution for under-representation (lower tail P-value) and asymptotic P-value for GSZ (see: Asymptotic P-value for GSZ score). All these scores were run with and without the inclusion of unknown sequences to the analysis.

The predicted output for regression was the Jaccard coefficient of similarity between the predicted GO class and the correct GO class nearest to the predicted GO class. The Jaccard coefficient of similarity was calculated by comparing the paths to the go classes via parental node paths until root. If the GO classes are identical, then the parental node paths will match perfectly, whereas if we compare different GO classes then the Jaccard coefficient of similarity depends on the proportion of common parental GO terms. Sparse regression was done using R packages Least Angle Regression (LARS) and Leaps as before (see chapter re-scoring sequence hits) using BLAST results for randomly selected 2,902 sequences with manually curated MF GO annotations. This time LARS provided more useful models. The selected variables were (1) GSZ with prior, (2) log of asymptotic GSZ P-value omitting unannotated sequences, and 3) log of hypergeometric P-value for under-representation. All were calculated by omitting the unannotated sequences from the analysis. The regression score (prediction for Jaccard coefficient of similarity) was used as input to the steps described below.

Generation of probability weights from GO class regression scores

The outcome from the earlier step should order the irrelevant GO classes and relevant GO classes. However, it does not provide any information on how reliable the obtained predictions are. Probability or reliability weights were a major requirement from the CAFA consortium. We solved this by using decision theory. The training set was identical to that used before. The steps were:

1. We define GO_{pos} = positive cases as GO class pairs with Jaccard coefficient of similarity > 0.9, and X_{pos} as corresponding regression scores;
2. We define GO_{neg} = negative cases as GO class pairs with Jaccard coefficient of similarity < 0.1, and X_{neg} as corresponding regression scores;
3. For every reported GO term we take its score as threshold, T;

Nature Methods: doi:10.1038/nmeth.2340

4. We define $P\{\text{smaller positive cases}\} = P(X_{pos} < T)$, the probability that a GO class from positive set has a smaller value than T;
5. We define $P\{\text{larger positive cases}\} = P(X_{neg} > T)$, the probability that a GO class from the negative set has a larger value than T;
6. We define probability weight $P_w = P(X_{pos} < T)/(P(X_{pos} < T) + P(X_{neg} > T))$.

P_w was used to give an estimate on the significance of the observed result.

Asymptotic P-value for GSZ score

For simplicity, the following text describes the case where the GSZ P-value is used to test the signal related to GO classes. It can be used more generally to monitor the signal related to any categories. Although GSZ has shown good performance, it has one weakness; it assumes similar Gaussian distribution in all cases. However, the sum of score values that GSZ monitors will depend on the number of GO class members observed in the SSRL. This number follows the hypergeometric distribution and its shape can be more similar to an exponential distribution for small GO classes.

One way to solve this problem is to generate a P-value estimating model that includes the hypergeometric probabilities. This model gives the probability for similar or larger sums at random. The generated model first uses the hypergeometric distribution to obtain the probability for every possible number of GO class members (N) and next it calculates $P(Z > Sum)$, the probability that the sum of regression scores X for N class members selected from the Sequence Search Result List (SSRL) is larger than the observed sum.

$$\text{P - value} = \sum_{N=N_{\min}}^{N_{\max}} P(N \mid M, L, K) \times P(Z > Sum \mid \mu_Z, \sigma_Z)$$

where M is the size of the class in the database, L is the size of the whole database, and K is the size of the SSRL. These represent the parameters required by the hypergeometric distribution. Next, the P-value for the sum is derived using cumulative normal distribution. We used for mean and standard deviation estimates represented in the GSZ article:

$$\mu_Z = N \times E(X)$$

and

$$\sigma_Z = \frac{N(K-N)}{K-1} Var(X)$$

where X defines the regression scores collected for the sequences in the $E(X)$ is the expectation value (mean) of the regression scores of sequences in the SSRL and $Var(X)$ defines the variance of regression scores. Note that these estimates depend on N (number of GO class members in SSRL). Overall, this P-value is a sum of combined weights, where the first weight comes from the hypergeometric distribution and the second weight from the cumulative normal distribution. A drawback of this method was a significantly larger running time than with GSZ. The

asymptotic P-values were used as input variables to the regression model used to select relevant GO classes.

Nature Methods: doi:10.1038/nmeth.2340

Tian Lab: A combined framework to annotate human gene functions using diverse types of omics data

Method

Overview of the prediction pipeline

In general, the data used to predict gene functions can be classified into the following three categories: the protein sequence encoded by the gene, the features associated with the gene (such as the phenotype of the gene knockout strain, the transcription factors regulating the gene, etc.), and the relationships between the gene and other genes in the genome (such as protein-protein interaction, gene expression correlation, genetic interaction, etc.). Because of the very different nature of these three categories of data, it is difficult to design a single model that uses all the data to predict the functions of the gene. In practice, it is often the case that only one type of data is used for the prediction. We have developed three independent algorithms in this study in order to use as much information as available about the gene, for the three categories of data in question: a protein sequence-based algorithm that uses PSI-BLAST output alignment to predict gene function (named psi-FSP), a gene feature-based algorithm that implements Random Forest to integrate diverse types of gene features to predict gene function (named GFP-RF), and a network-based algorithm that predicts gene function based on the topology of protein-protein interaction network (GFP-iPPI). These three algorithms are trained and applied independently to predict gene functions; a simple probabilistic model is used to combine their prediction results into the final prediction. Below we first describe these three algorithms separately. Then, we describe how to combine their prediction results.

Sequence-based function prediction algorithm – PSI-BLAST-based functional subtype prediction (psi-FSP)

Method description

To predict the function of a gene from its protein sequence, we first run a PSI-BLAST search with the protein sequence against the UniProtKB database to collect homologous sequences and prepare a PSI-BLAST-based multiple sequence alignment (MSA). PSI-BLAST is run with three iterations with default parameters. A query sequence-anchored multiple sequence alignment (MSA) with a maximal number of 20,000 sequence hits is obtained from the PSI-BLAST output. For sequences with more than one hit to the query sequence, only the most significant hit is included in the MSA. The pairwise sequence identity between all pairs of sequences in the MSA is computed, and the sequence hits with less than 15% sequence identity with the query sequence are removed from the MSA. In addition, for redundant sequences with above 90% sequence identity, the one with the smaller number or no GO annotations is removed from the MSA. The resultant filtered MSA is then used for predicting the functions of the query gene.

The GO annotations of all hit sequences in the PSI-BLAST-based MSA form a set of candidate functions for the query gene. To predict whether a gene is likely to have the function denoted by

Nature Methods: doi:10.1038/nmeth.2340

a given GO term *F*, we identify all hit sequences annotated with *F* in the PSI-BLAST-based MSA, and define the corresponding alignment as homo-functional MSA.

The whole PSI-BLAST-based MSA is considered as hetero-functional MSA, as it may include diverse types of functions. Then the goal is to compare the similarity of the query gene to the two MSAs. Rather than directly measuring the similarity using the whole MSA, we first identity the conserved positions in the MSA that can distinguish the homo-functional MSA from the hetero-functional MSA, and prepare a position-specific scoring matrix (PSSM) for *F*. Then, we compare the query sequence to the PSSM, and assign it a log-odds score, with a higher score indicating a higher probability for the query gene to have function *F*.

To determine the conserved positions, for every position *i* in the homo-functional MSA, we calculate the Shannon entropy as the follows:

$$H_{\text{homo}}(i) = -\sum_{AA\in\{\text{A, C, ..., Y}\}} p(i, AA)\log p(i, AA)$$

in which *AA* is one of 20 amino acids, and $p(i, AA)$ is the frequency of the amino acid *AA* at site *i* among the whole homo-functional group, defined as N_{AA}/N, where N_{AA} is the total number of *AA* at site *i*, and *N* is the total number of sequences in the homo-functional MSA. Particularly, when a gap ("-") is encountered when counting N_{AA}, then 1/20 will be added to N_{AA}. The entropy score is converted to a conservation score as the follows:

$$C_{\text{homo}}(i) = e^{-H_{\text{homo}}(i)}$$

with $C_{\text{homo}}(i) = 1$ indicating complete conservation. For hetero-functional MSA, we calculate two types of entropy at each *AA* site: $H_{\text{hetero}}(i)$ and $H_{\text{hetero\_AAgroup}}(i)$, and convert them into conservation scores $C_{\text{hetero}}(i)$ and $C_{\text{hetero\_AAgroup}}(i)$, respectively. $H_{\text{hetero}}(i)$ is calculated using the same formula as $H_{\text{homo}}(i)$, while $H_{\text{hetero\_AAgroup}}(i)$ is calculated in a similar way except that 10 classes of *AA*s instead of 20 kinds of AA are used. The classification of *AA* is based on the amino acid's substitution rate table: Group 1: Ala; Group 2: Cys; Group 3: Asp and Glu; Group 4: Phe, Trp and Tyr; Group 5: Gly; Group 6: His and Asn; Group 7: Ile, Leu, Meu, and Val; Group 8: Pro; Group 9: Lys, Gln, and Arg; Group 10: Ser and Thr. Then, the conservation score for each site *i* is calculated as the follows:

$$C(i) = C_{\text{homo}}(i) + C_{\text{hetero}}(i) - 0.5\cdot C_{\text{hetero\_AAgroup}}(i)$$

Higher $C(i)$ implies that site *i* is more conserved at amino acid level in the homo-functional MSA, and more conserved at amino acid group level while relatively less conserved at amino acid level in the hetero-functional MSA. Thus, it conveys more functional sub-type specificity information, and makes it easier to distinguish function *F* from other functions in the hetero-functional MSA.

We perform a Z-transformation for the final conservation score and select sites with normalized scores greater than 1 to prepare the PSSM for function *F*. The PSSM consists of log-odds of the observed frequency to the background frequency of each amino acid *AA* at every selected conserved site. The log-odds of an amino acid *AA* at site *i* is calculated as follows:

Nature Methods: doi:10.1038/nmeth.2340

$$\log\text{-odds}(AA,i) = \frac{Freq_{\text{homo}}(AA,i) + Freq(AA,bg)/N_{\text{homo}}}{Freq_{\text{hetero}}(AA,i) + Freq(AA,bg)/N_{\text{hetero}}}$$

where $Freq_{\text{homo}}(AA,i)$ and $Freq_{\text{hetero}}(AA,i)$ are the frequency of *AA* at site *i* in the homo-functional and hetero-functional MSA, respectively, $Freq(AA,bg)$ is the background frequency of *AA* in the whole MSA, and N_{homo} and N_{hetero} are the number of sequences in the homo-functional MSA and hetero-functional MSA, respectively. $Freq(AA,bg)$ is used to avoid an infinite log-odds score. Finally, the prediction score of a query gene for GO term *F* is the sum of all log-odds scores over the selected sites:

$$Score(F) = \sum_{i=1}^{n} \log\text{-odds}(AA,i)$$

where *n* is the total number of conserved sites, and *AA* is the amino acid of query gene at the *i*-th conserved site. The higher the score, the more likely the query gene is to have function *F*.

Benchmark of psi-FSP

We download the GO terms and the UniProtKB[97] gene annotations on 2012-2-1. GO annotations with evidence code "IEA" or "RCA" are excluded from our analysis. Then, we randomly select 10,000 proteins that have been annotated with at least one GO term in UniRef50[98] (released in Dec 2010), which have less than 50% sequence identity to each other. We then apply the psi-FSP method to predict the functions for each of the selected proteins, and compare the prediction results with the original annotations using a Precision-Recall curve. To further evaluate the performance of psi-FSP, we use the E-value and the sequence identity of the PSI-Blast output to predict the functions of the selected proteins. The E-value-based method is applied as follows: the smallest E-value among the sequences in the homo-functional MSA of GO term *F* is used as the prediction score of *F* for the query protein. The sequence-identity-based method is similar to the E-value-based method, except that the maximal sequence identity of the proteins to the query protein in the homo-functional MSA is reported as the prediction score.

Application of psi-FSP to predict human gene function

The gene annotations for 19,937 protein-coding-genes in the human genome are obtained from the Ensembl Gene version 59.[99] Each gene is searched with PSI-BLAST against the UniProtKB database with three iterations and default parameters. After filtering the query-anchored PSI-BLAST output MSA, the psi-FSP method is applied to predict the functions of human genes.

Nature Methods: doi:10.1038/nmeth.2340

Feature-based function prediction algorithms – Gene Function Prediction using Random Forest (GFP-RF)

Data collection and feature construction

The gene annotations for 19,937 protein-coding-genes are obtained from the Ensembl Gene database, version 59. Protein domain features of the human genes are obtained from the UniProt database,[100] which includes protein domain information from ten protein domain databases (InterPro,[101] Pfam,[102] TIGRFAMs,[103] SUPFAM,[104] SMART,[105] PROSITE,[106] PRINTS,[107] PANTHER,[108] HAMAP[109] and Gene3D[110]) and DrugBank.[111] As for regulatory related information, we download histone modifications, histone variants, chromatin structures, repeat elements (aligned from sequence given by RepBase[112]) and predicted miRNA targets (originally from the miRanda database[113]) from Ensembl, while the transcription factor binding sites (TFBS) are acquired from the UCSC genome browser[114] with two tracks: experimentally confirmed TFBS by ChIP-Seq experiments from track wgEncodeRegTfbsClustered (generate by ENCODE[115]) and the predicted TFBS from track tfbsConsSites (alignment from position weighted matrix from TRANSFAC 7.0[116]).

From the downloaded data, we construct seven types of gene features: protein domain, drug target, experimental TFBS, predicted TFBS, histone modifications, miRNA binding sites and repeat elements. The histone modifications feature also combines open chromatin feature DNase I hypersensitive sites, (DHS). In the construction of features, protein domain and drug target features are treated as binary attributes, where the attribute equals 1 if the protein is annotated with the feature and 0 otherwise. For the features that are annotated by chromosome positions, a gene is annotated with the feature if it is located within a given region of a gene. Our regions-of-interests include the promoter region (defined as 1500bp on both sides of the TSS in this paper) for experimental TFBS, predicted TFBS and histone modifications features and the 3' region (defined as 1500bp upstream of 3' end to the 3' end of the gene) for miRNA binding feature. Because the regulatory features may have cell type specificity, for the experimental regulatory features, namely the experimental TFBS and histone modification, the feature together with its cell line of the experiment is considered as one attribute.

Gene Function Prediction using Random Forest (GFP-RF)

A slightly modified random forest (RF) algorithm, described by Tian et al.[117], is used to annotate the GO terms of unknown genes from the constructed gene features. Briefly, for every GO term, we construct a RF model that includes an ensemble of decision trees. Each decision tree uses a bootstrapped sample set to train the decision tree model and gives a prediction score for genes that are not used in training the model. At every branch node of the decision tree, only a small subset of attributes (gene features) are randomly selected from the attribute pool, and are evaluated to find the best attributes. A probability score is then given at a leaf node by dividing the number of true positives (genes annotated with the GO term) to the total number of genes. The final prediction score of a given gene is the averaged probability score of the gene across the decision trees in which the gene is not used in training the model. Different from the traditional RF in which every decision tree is fully grown, in the modified RF, after the bootstrapping step we conduct feature selection before constructing the decision trees by evaluating the

Nature Methods: doi:10.1038/nmeth.2340

hypergeometric distribution probability of a feature, and remove features with the probability above 0.01. In addition, we implement the early-stopping criteria when constructing the decision tree, and similarly use the hypergeometric distribution probability of 0.01 to determine whether to stop growing the tree. For each RF model, we generate 200 decision trees. To evaluate the performance of the RF in predicting gene functions, a precision/recall curve is plotted using the prediction scores of all genes. To compare the prediction power of each of the seven types of gene features, we also run the RF model with each type of gene features independently to 1 predict gene functions.

Network-based function prediction – Gene Function Prediction using an Integrated Protein-Protein Interaction Network (GFP-iPPI)

We construct a human protein-protein interaction (PPI) network from the union of downloaded protein-protein interaction data obtained from Biogrid,[118] MINT[119] and IntAct[120] databases. The PPI network is used to integrate the gene features for predicting gene functions. Our network-based algorithm relies on a modified neighborhood counting algorithm, which calculates a χ^2 like score following the formula:

$$Score(p, f) = \frac{(n_f - e_f)^2}{e_f}$$

where

$$n_f = \sum_{i \in \text{neighbor}} w_i x_i .$$

Here, p stands for a protein, f stands for a function to predict, n_f is the summed weighted degree of p with proteins in the direct neighbors of p that have the function f and e_f is the expected weighted degree of proteins with function f in the network. In the calculation of n_f, the x_i is 1 if the neighbor is annotated with the given GO term and otherwise 0; w_i is the weight of the link between protein p and protein x_i. For the raw PPI network, w_i is equal to 1. Additionally, we apply the information of gene features to assign weights to the edge of PPI network using Jaccard Similarity Coefficient (JSC) between the connected two nodes. Here, w_i is defined as the ratio of the number of common sharing features to the total number of features from the two genes, which ranges from 0 to 1. To evaluate the prediction performance, a precision/recall curve is plotted using the prediction scores. To compare the usefulness of each type of gene features, JSC is calculated based on each type of gene features and apply to the modified neighborhood counting algorithm independently.

Nature Methods: doi:10.1038/nmeth.2340

Combining the predictions results from psi-FSP, GFP-RF, and GFP-iPPI

Conversion of prediction scores to precision scores

Each of the three above-described algorithms gives prediction scores in different ranges and different distributions. For example, the prediction scores by psi-FSP are log-odds scores ranging from minus to plus large scores; the prediction scores by GFP-RF are probability scores, ranging from 0 to 1; the prediction scores by GFP-iPPI are chi-square scores, ranging from 0 to plus infinity. Thus, it is difficult to directly combine the prediction scores from the three algorithms using a single model. To solve this problem we plot a precision/recall curve for the prediction scores given by each algorithm and use the precision/recall curve to convert prediction scores into precision scores that range from 0 to 1. As each point of the precision/recall curve corresponds to the precision and recall computed from a prediction score cut-off, this prediction score cut-off is assigned the corresponding precision as the converted score, termed "precision score" here. Because of the lack of enough data points to calculate the precision and recall, sometimes the precision score corresponding to a higher prediction score may be smaller than that corresponding to a lower prediction score. In that case, we update the precision score corresponding to the higher prediction score with the precision score corresponding to the lower prediction score. For the prediction scores of psi-FSP, we use all the prediction scores to plot the precision/recall curve and transform them into precision scores. Because the 1 prediction scores are largely dependent on the number of genes associated with the GO term in the human genome for GFP-RF and GFP-iPPI, we divide the GO terms into different categories according to the number of associated genes (from 3 to 500 genes), and plot a precision/recall curve for each category of GO predictions in order to obtain the precision scores. By converting the prediction scores into precision scores, the precision scores given by each algorithm are now comparable and can be combined to generate the final prediction scores.

A simple probabilistic model to combine the precision scores made by the three algorithms

Because different types of gene features are used independently in GFP-RF and GFP-iPPI to predict gene functions, we first combine the precision scores generated by different types of gene features with these two algorithms separately. To do so, for a given gene-GO pair, we use the joint probability to combine the precision scores (a gene-GO pair), given by

$$P = 1 - \prod_{i \in C} (1 - p_i)$$

in which i represents the type of data, C represents all types of data that have an precision score greater than 0.1, and p_i is the precision score obtained from data type i. Thus, we obtain a precision score for a given gene-GO pair from each of the three algorithms. Then, in a similar way, the three precision scores are combined to generate the final prediction score.

Nature Methods: doi:10.1038/nmeth.2340

References

1. Liolios, K. et al. The Genomes On Line Database (GOLD) in 2009: status of genomic and metagenomic projects and their associated metadata. *Nucleic Acids Res* 38, D346-354 (2010).
2. Bork, P. et al. Predicting function: from genes to genomes and back. *J Mol Biol* 283, 707-725 (1998).
3. Rost, B., Liu, J., Nair, R., Wrzeszczynski, K.O. & Ofran, Y. Automatic prediction of protein function. *Cell Mol Life Sci* 60, 2637-2650 (2003).
4. Watson, J.D., Laskowski, R.A. & Thornton, J.M. Predicting protein function from sequence and structural data. *Curr Opin Struct Biol* 15, 275-284 (2005).
5. Friedberg, I. Automated protein function prediction--the genomic challenge. *Brief Bioinform* 7, 225-242 (2006).
6. Sharan, R., Ulitsky, I. & Shamir, R. Network-based prediction of protein function. *Mol Syst Biol* 3, 88 (2007).
7. Lee, D., Redfern, O. & Orengo, C. Predicting protein function from sequence and structure. *Nat Rev Mol Cell Biol* 8, 995-1005 (2007).
8. Punta, M. & Ofran, Y. The rough guide to in silico function prediction, or how to use sequence and structure information to predict protein function. *PLoS Comput Biol* 4, e1000160 (2008).
9. Rentzsch, R. & Orengo, C.A. Protein function prediction--the power of multiplicity. *Trends Biotechnol* 27, 210-219 (2009).
10. Xin, F. & Radivojac, P. Computational methods for identification of functional residues in protein structures. *Curr Protein Pept Sci* 12, 456-469 (2011).
11. Altschul, S.F. et al. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res* 25, 3389-3402 (1997).
12. Jensen, L.J. et al. Prediction of human protein function from post-translational modifications and localization features. *J Mol Biol* 319, 1257-1265. (2002).
13. Wass, M.N. & Sternberg, M.J. ConFunc--functional annotation in the twilight zone. *Bioinformatics* 24, 798-806 (2008).
14. 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).
15. Hawkins, T., Luban, S. & Kihara, D. Enhanced automated function prediction using distantly related sequences and contextual association by PFP. *Protein Sci* 15, 1550-1556 (2006).
16. Clark, W.T. & Radivojac, P. Analysis of protein function and its prediction from amino acid sequence. *Proteins* 79, 2086-2096 (2011).
17. Pellegrini, M., Marcotte, E.M., Thompson, M.J., Eisenberg, D. & Yeates, T.O. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *Proc Natl Acad Sci USA* 96, 4285-4288 (1999).
18. Marcotte, E.M. et al. Detecting protein function and protein-protein interactions from genome sequences. *Science* 285, 751-753 (1999).
19. Enault, F., Suhre, K. & Claverie, J.M. Phydbac "Gene Function Predictor": a gene annotation tool based on genomic context analysis. *BMC Bioinformatics* 6, 247 (2005).
20. Engelhardt, B.E., Jordan, M.I., Muratore, K.E. & Brenner, S.E. Protein molecular function prediction by Bayesian phylogenomics. *PLoS Comput Biol* 1, e45 (2005).

Nature Methods: doi:10.1038/nmeth.2340

21. Gaudet, P., Livstone, M.S., Lewis, S.E. & Thomas, P.D. Phylogenetic-based propagation of functional annotations within the Gene Ontology consortium. *Brief Bioinform* 12, 449-462 (2011).
22. Deng, M., Zhang, K., Mehta, S., Chen, T. & Sun, F. Prediction of protein function using protein-protein interaction data. *J Comput Biol* 10, 947-960 (2003).
23. Letovsky, S. & Kasif, S. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics* 19 Suppl 1, i197-204 (2003).
24. Vazquez, A., Flammini, A., Maritan, A. & Vespignani, A. Global protein function prediction from protein-protein interaction networks. *Nat Biotechnol* 21, 697-700 (2003).
25. Nabieva, E., Jim, K., Agarwal, A., Chazelle, B. & Singh, M. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics* 21 Suppl 1, i302-310 (2005).
26. Pazos, F. & Sternberg, M.J. Automated prediction of protein function and detection of functional sites from structure. *Proc Natl Acad Sci USA* 101, 14754-14759 (2004).
27. Pal, D. & Eisenberg, D. Inference of protein function from protein structure. *Structure* 13, 121-130 (2005).
28. Laskowski, R.A., Watson, J.D. & Thornton, J.M. Protein function prediction using local 3D templates. *J Mol Biol* 351, 614-626 (2005).
29. Huttenhower, C., Hibbs, M., Myers, C. & Troyanskaya, O.G. A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics* 22, 2890-2897 (2006).
30. Troyanskaya, O.G., Dolinski, K., Owen, A.B., Altman, R.B. & Botstein, D. A Bayesian framework for combining heterogeneous data sources for gene function prediction (in Saccharomyces cerevisiae). *Proc Natl Acad Sci USA* 100, 8348-8353 (2003).
31. Lee, I., Date, S.V., Adai, A.T. & Marcotte, E.M. A probabilistic functional network of yeast genes. *Science* 306, 1555-1558 (2004).
32. Costello, J.C. et al. Gene networks in Drosophila melanogaster: integrating experimental data to predict gene function. *Genome Biol* 10, R97 (2009).
33. Kourmpetis, Y.A., van Dijk, A.D., Bink, M.C., van Ham, R.C. & ter Braak, C.J. Bayesian Markov Random Field analysis for protein function prediction based on network data. *PLoS One* 5, e9293 (2010).
34. Sokolov, A. & Ben-Hur, A. Hierarchical classification of gene ontology terms using the GOstruct method. *J Bioinform Comput Biol* 8, 357-376 (2010).
35. Ashburner, M. et al. Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet* 25, 25-29 (2000).
36. Bairoch, A. et al. The Universal Protein Resource (UniProt). *Nucleic Acids Res* 33 Database Issue, D154-159 (2005).
37. Schnoes, A.M., Brown, S.D., Dodevski, I. & Babbitt, P.C. Annotation error in public databases: misannotation of molecular function in enzyme superfamilies. *PLoS Comput Biol* 5, e1000605 (2009).
38. Punta, M. et al. The Pfam protein families database. *Nucleic Acids Res* 40, D290-301 (2012).
39. Wang, G. et al. PNPASE regulates RNA import into mitochondria. *Cell* 142, 456-467 (2010).

Nature Methods: doi:10.1038/nmeth.2340

40. Sarkar, D. et al. Down-regulation of Myc as a potential target for growth arrest induced by human polynucleotide phosphorylase (hPNPaseold-35) in human melanoma cells. *J Biol Chem* 278, 24542-24551 (2003).
41. Wu, J. & Li, Z. Human polynucleotide phosphorylase reduces oxidative RNA damage and protects HeLa cell against oxidative stress. *Biochem Biophys Res Commun* 372, 288-292 (2008).
42. Wang, D.D., Shu, Z., Lieser, S.A., Chen, P.L. & Lee, W.H. Human mitochondrial SUV3 and polynucleotide phosphorylase form a 330-kDa heteropentamer to cooperatively degrade double-stranded RNA with a 3'-to-5' directionality. *J Biol Chem* 284, 20812-20821 (2009).
43. Portnoy, V., Palnizky, G., Yehudai-Resheff, S., Glaser, F. & Schuster, G. Analysis of the human polynucleotide phosphorylase (PNPase) reveals differences in RNA binding and response to phosphate compared to its bacterial and chloroplast counterparts. *RNA* 14, 297-309 (2008).
44. Jeffery, C.J. Moonlighting proteins. *Trends Biochem Sci* 24, 8-11 (1999).
45. Khersonsky, O. & Tawfik, D.S. Enzyme promiscuity: a mechanistic and evolutionary perspective. *Annu Rev Biochem* 79, 471-505 (2010).
46. Brenner, S.E. Errors in genome annotation. *Trends Genet* 15, 132-133 (1999).
47. Doolittle, R.F. Of URFS and ORFS: A Primer on How to Analyze Derived Amino Acid Sequences. (University Science Books, 1986).
48. Addou, S., Rentzsch, R., Lee, D. & Orengo, C.A. Domain-based and family-specific sequence identity thresholds increase the levels of reliable protein function transfer. *J Mol Biol* 387, 416-430 (2009).
49. Nehrt, N.L., Clark, W.T., Radivojac, P. & Hahn, M.W. Testing the ortholog conjecture with comparative functional genomic data from mammals. *PLoS Comput Biol* 7, e1002073 (2011).
50. Brown, S.D., Gerlt, J.A., Seffernick, J.L. & Babbitt, P.C. A gold standard set of mechanistically diverse enzyme superfamilies. *Genome Biol* 7, R8 (2006).
51. Gerlt, J.A. et al. The Enzyme Function Initiative. *Biochemistry* 50, 9950-9962 (2011).
52. Barrell, D. et al. The GOA database in 2009--an integrated Gene Ontology Annotation resource. *Nucleic Acids Res* 37, D396-403 (2009).
53. Hanley, J. & McNeil, B.J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology* 143, 29-36 (1982).
54. Cozzeto, D., Buchan, D.W.A., Bryson, K. & Jones, D.T. Protein function prediction by massive integration of evolutionary analyses and multiple data sources. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S1.
55. Falda, M. et al. Argot2: a large scale function prediction tool relying on semantic similarity of weighted Gene Ontology terms. *BMC Bioinformatics* 13, S14 (2012).
56. Chitale, M., Hawkins, T., Park, C. & Kihara, D. ESG: extended similarity group method for automated protein function prediction. *Bioinformatics* 25, 1739-1745 (2009).
57. Bartoli, L. et al. The Bologna Annotation Resource: a non hierarchical method for the functional and structural annotation of protein sequences relying on a comparative large-scale genome analysis. *J Proteome Res* 8, 4362-4371 (2009).
58. Piovesan, D. et al. BAR-PLUS: the Bologna Annotation Resource Plus for functional and structural annotation of protein sequences. *Nucleic Acids Res* 39, W197-202 (2011).

Nature Methods: doi:10.1038/nmeth.2340

59. Wang, Z., Cao, R. & Cheng, J. Three-level prediction of protein function by combining profile-sequence search, profile-profile search, and domain co-occurrence networks. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S3.
60. Rentzsch, R. & Orengo, C.A. Protein function prediction using domain families. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S5.
61. Erdin, S., Venner, E., Lisewski, M.A. & Lichtarge, O. Function prediction from networks of local evolutionary similarity in protein structure. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S6.
62. Engelhardt, B.E., Jordan, M.I., Srouji, J.R. & Brenner, S.E. Genome-scale phylogenetic function annotation of large and diverse protein families. *Genome Res* 21, 1969-1980 (2011).
63. Fang, H. & Gough, J. dcGO: database of domain-centric ontologies on functions, phenotypes, diseases and more. *Nucleic Acids Res* (2012).
64. Fang, H. & Gough, J. A domain-centric solution to functional genomics via dcGO Predictor. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S9.
65. Lan, L., Djuric, N., Guo, Y. & Vucetic, S. MS-kNN: Protein function prediction by integrating multiple data sources. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S9.
66. Hamp, T. et al. Homology-based inference sets the bar high for protein function prediction. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S7.
67. Kourmpetis, Y.A., van Dijk, A.D., van Ham, R.C. & ter Braak, C.J. Genome-wide computational function prediction of Arabidopsis proteins by integration of multiple data sources. *Plant Physiol* 155, 271-281 (2011).
68. Wong, A. & Shatkay, H. Protein function prediction using text-based features extracted from the biomedical literature: the CAFA challenge. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S14.
69. Lobley, A., Swindells, M.B., Orengo, C.A. & Jones, D.T. Inferring function using patterns of native disorder in proteins. *PLoS Comput Biol* 3, e162 (2007).
70. Lobley, A.E., Nugent, T., Orengo, C.A. & Jones, D.T. FFPred: an integrated feature-based function prediction server for vertebrate proteomes. *Nucleic Acids Res* 36, W297-302 (2008).
71. Hawkins, T., Chitale, M., Luban, S. & Kihara, D. PFP: Automated prediction of gene ontology functional annotations with confidence scores using protein sequence data. *Proteins* 74, 566-582 (2009).
72. de Lima Morais, D.A. et al. SUPERFAMILY 1.75 including a domain-centric gene ontology method. *Nucleic Acids Res* 39, D427-434 (2011).
73. Weinhold, N., Sander, O., Domingues, F.S., Lengauer, T. & Sommer, I. Local function conservation in sequence and structure space. *PLoS Comput Biol* 4, e1000105 (2008).
74. Fontana, P., Cestaro, A., Velasco, R., Formentin, E. & Toppo, S. Rapid annotation of anonymous sequences from genome projects using semantic similarities and a weighting scheme in gene ontology. *PLoS One* 4, e4619 (2009).
75. Schietgat, L. et al. Predicting gene function using hierarchical multi-label decision tree ensembles. *BMC Bioinformatics* 11, 2 (2010).
76. Sokolov, A., Funk, C., Graim, K., Verspoor, K. & Ben-Hur, A. Combining heterogeneous data sources for accurate functional annotation of proteins. *BMC Bioinformatics* (2013), doi:10.1186/1471-2105-14-S3-S10.

77. Toronen, P., Ojala, P.J., Marttinen, P. & Holm, L. Robust extraction of functional signals from gene set analysis using a generalized threshold free scoring function. *BMC Bioinformatics* 10, 307 (2009).
78. Conesa, A. et al. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics* 21, 3674-3676 (2005).
79. Arakaki, A.K., Huang, Y. & Skolnick, J. EFICAz2: enzyme function inference by a combined approach enhanced by machine learning. *BMC Bioinformatics* 10, 107 (2009).
80. Camon, E.B. et al. An evaluation of GO annotation retrieval for BioCreAtIvE and GOA. *BMC Bioinformatics* 6 Suppl 1, S17 (2005).
81. Lichtarge, O., Bourne, H.R. & Cohen, F.E. An evolutionary trace method defines binding surfaces common to protein families. *J. Mol. Biol.* 257, 342-358 (1996).
82. Erdin, S., Lisewski, A.M. & Lichtarge, O. Protein function prediction: towards integration of similarity metrics. *Curr Opin Struct Biol* 21, 180-188 (2011).
83. Kristensen, D.M. et al. Prediction of enzyme function based on 3D templates of evolutionarily important amino acids. *BMC Bioinformatics* 9, 17 (2008).
84. Ward, R.M. et al. De-orphaning the structural proteome through reciprocal comparison of evolutionarily important structural features. *PLoS One* 3, e2136 (2008).
85. Erdin, S., Ward, R.M., Venner, E. & Lichtarge, O. Evolutionary trace annotation of protein function in the structural proteome. *J Mol Biol* 396, 1451-1473 (2010).
86. Venner, E. et al. Accurate protein structure annotation through competitive diffusion of enzymatic functions over a network of local evolutionary similarities. *PLoS One* 5, e14286 (2010).
87. von Mering, C. et al. STRING: known and predicted protein-protein associations, integrated and transferred across organisms. *Nucleic Acids Res* 33, D433-437 (2005).
88. Cuff, A.L. et al. Extending CATH: increasing coverage of the protein structure universe and linking structure with function. *Nucleic Acids Res* 39, D420-426 (2011).
89. Lee, D.A., Rentzsch, R. & Orengo, C. GeMMA: functional subfamily classification within superfamilies of predicted protein structural domains. *Nucleic Acids Res* 38, 720-737 (2010).
90. Lees, J., Yeats, C., Redfern, O., Clegg, A. & Orengo, C. Gene3D: merging structure and function for a Thousand genomes. *Nucleic Acids Res* 38, D296-300 (2010).
91. Zhou, D., Bousquet, O., Navin Lal, T., Weston, J. & Schölkopf, B. Learning with local and global consistency. *Advances in Neural Information Processing Systems*, 321-328 (2004).
92. Tsochantaridis, I., Joachims, T., Hofmann, T. & Altun, Y. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* 8, 1453-1484 (2005).
93. Bjorne, J., Ginter, F., Pyysalo, S., Tsujii, J. & Salakoski, T. Complex event extraction at PubMed scale. *Bioinformatics* 26, i382-390 (2010).
94. Brady, S. & Shatkay, H. EpiLoc: a (working) text-based system for predicting protein subcellular location. *Pac Symp Biocomput*, 604-615 (2008).
95. Eddy, S.R. A new generation of homology search tools based on probabilistic inference. *Genome Inform* 23, 205-211 (2009).
96. Kuzniar, A. et al. ProGMap: an integrated annotation resource for protein orthology. *Nucleic Acids Res* 37, W428-434 (2009).
97. Harris, M.A. et al. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Res* 32, D258-261 (2004).

Nature Methods: doi:10.1038/nmeth.2340

98. Suzek, B.E., Huang, H., McGarvey, P., Mazumder, R. & Wu, C.H. UniRef: comprehensive and non-redundant UniProt reference clusters. *Bioinformatics* 23, 1282-1288 (2007).
99. Hubbard, T. et al. The Ensembl genome database project. *Nucleic Acids Res* 30, 38-41 (2002).
100. Wu, C.H. et al. The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Res* 34, D187-191 (2006).
101. Apweiler, R. et al. The InterPro database, an integrated documentation resource for protein families, domains and functional sites. *Nucleic Acids Res.* 29, 37-40 (2001).
102. Bateman, A. et al. The Pfam protein families database. *Nucleic Acids Res.* 32, 138-141 (2004).
103. Haft, D.H., Selengut, J.D. & White, O. The TIGRFAMs database of protein families. *Nucleic Acids Res.* 31, 371-373 (2003).
104. Pandit, S.B. et al. SUPFAM--a database of potential protein superfamily relationships derived by comparing sequence-based and structure-based families: implications for structural genomics and function annotation in genomes. *Nucleic Acids Res* 30, 289-293 (2002).
105. Schultz, J., Copley, R.R., Doerks, T., Ponting, C.P. & Bork, P. SMART: a web-based tool for the study of genetically mobile domains. *Nucleic Acids Res* 28, 231-234 (2000).
106. Hofmann, K., Bucher, P., Falquet, L. & Bairoch, A. The PROSITE database, its status in 1999. *Nucleic Acids Res.* 27, 215-219 (1999).
107. Attwood, T.K. The PRINTS database: a resource for identification of protein families. *Brief Bioinform* 3, 252-263 (2002).
108. Thomas, P.D. et al. PANTHER: a library of protein families and subfamilies indexed by function. *Genome Res* 13, 2129-2141 (2003).
109. Lima, T. et al. HAMAP: a database of completely sequenced microbial proteome sets and manually curated microbial protein families in UniProtKB/Swiss-Prot. *Nucleic Acids Res* 37, D471-478 (2009).
110. Buchan, D.W. et al. Gene3D: structural assignments for the biologist and bioinformaticist alike. *Nucleic Acids Res* 31, 469-473 (2003).
111. Wishart, D.S. et al. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucleic Acids Res* 34, D668-672 (2006).
112. Kohany, O., Gentles, A.J., Hankus, L. & Jurka, J. Annotation, submission and screening of repetitive elements in Repbase: RepbaseSubmitter and Censor. *BMC Bioinformatics* 7, 474 (2006).
113. John, B. et al. Human MicroRNA targets. *PLoS Biol* 2, e363 (2004).
114. Karolchik, D. et al. The UCSC Genome Browser Database. *Nucleic Acids Res* 31, 51-54 (2003).
115. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science* 306, 636-640 (2004).
116. Matys, V. et al. TRANSFAC and its module TRANSCompel: transcriptional gene regulation in eukaryotes. *Nucleic Acids Res* 34, D108-110 (2006).
117. Tian, W. et al. Combining guilt-by-association and guilt-by-profiling to predict Saccharomyces cerevisiae gene function. *Genome Biol* 9 Suppl 1, S7 (2008).
118. Stark, C. et al. BioGRID: a general repository for interaction datasets. *Nucleic Acids Res* 34, D535-539 (2006).
119. Zanzoni, A. et al. MINT: a Molecular INTeraction database. *FEBS Lett* 513, 135-140. (2002).

Nature Methods: doi:10.1038/nmeth.2340

120. Hermjakob, H. et al. IntAct: an open source molecular interaction database. *Nucleic Acids Res* 32, D452-455 (2004).

Nature Methods: doi:10.1038/nmeth.2340