THE FAT-CAT WEB SERVER

A BRIEF TUTORIAL

Last revised 2/11/2013

Look for this icon to find helpful hints on using FAT-CAT and interpreting results.

Figures

1. PHYLOGENOMICS, PHYLOFACTS AND FAT-CAT

1.1 The raison d'etre: problems with standard annotation transfer approaches

Systematic errors in gene functional annotation. The standard protocol to functional annotation uses an approach is called "*annotation transfer*" or "*transitive annotation.*" The process involves using BLAST to find a homolog followed by transferring the annotation of the homolog to the query. Unfortunately, this approach is now known to be prone to systematic error [2-4]due to gene duplication, domain rearrangements and existing errors in annotation [3, 5-9]; these sources of annotation error are illustrated in Figure 1. A recent study of six manually curated enzyme superfamilies shows functional annotation error rates ranging from 5%-63% in the major sequence databases and KEGG [10]. Annotation transfer protocols also propagate existing annotation errors [6, 11].

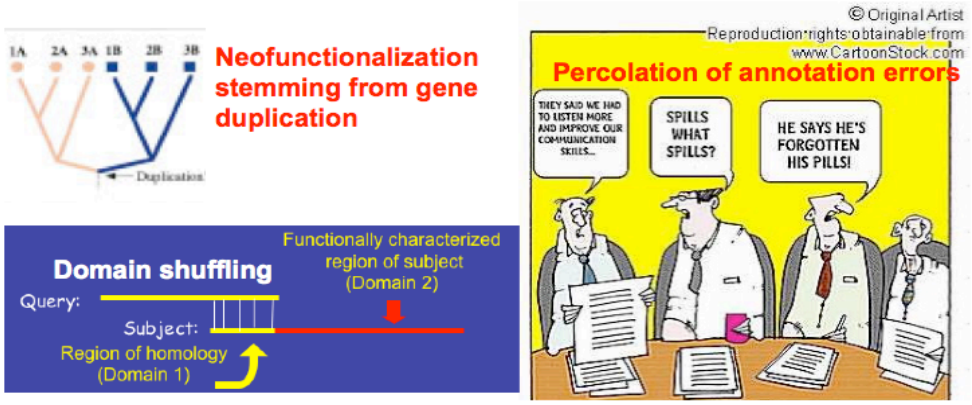


Figure 1. Sources of functional annotation error resulting from annotation transfer protocols. *Top left*: Gene duplication produces paralogous genes, which can acquire novel functions (*neofunctionalization*) or partition the ancestral function (*subfunctionalization*). *Bottom left*: Domain shuffling modifies domain architecture. In this example, transferring the annotation of the database hit (subject) to the query would lead to an error. *Right*: Percolation of existing annotation errors.

1.2 What is meant by phylogenomic prediction of function?

Protein families evolve a multiplicity of functions through gene duplication, speciation and other processes. As a number of studies have shown, standard methods of protein function prediction produce systematic errors on these data.

Phylogenomic analysis—combining phylogenetic tree construction, integration of experimental data and differentiation of orthologs and paralogs—has been proposed to address these errors and improve the accuracy of functional classification [1, 12].

The term *phylogenomics* was proposed initially by Eisen to describe the use of phylogenetic analysis to improve the accuracy of gene functional annotation [1]; it is also used to describe species phylogeny estimation using multiple genes (e.g., as in a concatenated gene matrix approach) [13]. A related approach was developed for the functional annotation of the human genome [14], using the SCI-PHY algorithm [15] to identify functional subfamilies, and subsequently extended into two phylogenomic databases of gene family phylogenies: the Panther tools [16] and the PhyloFacts resource [17, 18].

The explicit integration of structure prediction and analysis in this framework, which we call structural phylogenomics, provides additional insights into protein superfamily evolution [19].

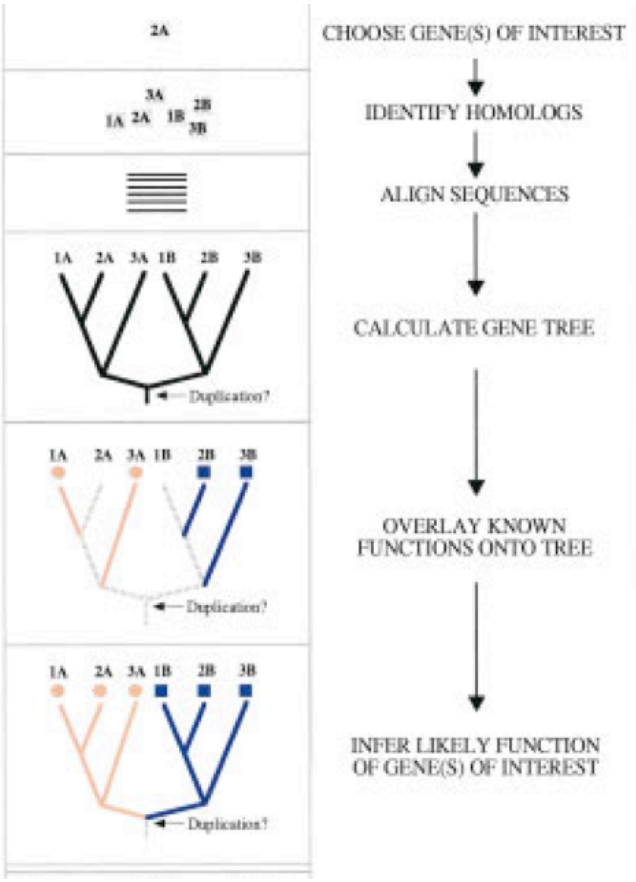


Figure 2. The phylogenomic analysis pipeline [1].

1.3 Orthologs, super-orthologs, paralogs and other homology relationships

The term *ortholog* was first proposed by Walter Fitch [20] to differentiate genes related by speciation from those related by duplication events : "Where the homology is the result of gene duplication so that both copies have descended side by side during the history of an organism…the genes should be called *paralogous* (para = in parallel). Where the homology is the result of speciation so that the history of the gene reflects the history of the species…the genes should be called *orthologous* (ortho = exact)."

Note that orthology is a phylogenetic term, but is used in practice as a surrogate for functional equivalence: in fact, orthologs in distantly related species may have diverged functionally from their common ancestor.

Because orthology is not transitive (that is, if X and Y are orthologs, and Y and Z are orthologs, it does not necessarily follow that X and Z are orthologs) [21], Zmasek and Eddy [22] proposed a more restrictive definition of orthology that explicitly disallows any duplication events: two genes X and Y are *super-orthologs* if and only if every node on the evolutionary tree relating them corresponds to a speciation event. The super-orthology relation has the advantage of being transitive as it partitions the gene family tree into super-orthologous subtrees. Sonnhammer and Koonin developed related terms to describe in-species duplication events (called *inparalogs*) and other types of paralogy [23]. Orthology, super-orthology and inparalog relationships are illustrated below.

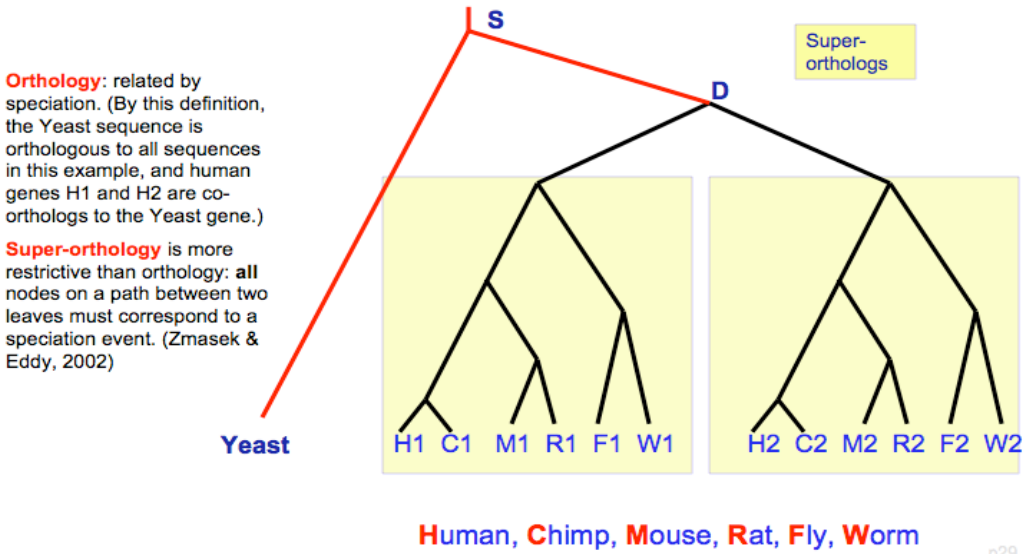


Figure 3. Orthology definitions. Two sequences are orthologs if they are related by a speciation event. Two sequences are super-orthologs if every node on the path in the tree joining them corresponds to a speciation event. Note that super-orthology is transitive while the standard definition of orthology is not.

In this example, the yeast sequence is related by a speciation event (S) to every sequence in the tree and is thus orthologous to all, but is not super-orthologous to any. However, within each boxed subtree there are no intervening duplication events and sequences in each of these individual subtrees are super-orthologous. Sequences in the left boxed subtree are all paralogous to sequences in the right boxed subtree.

Super-orthology allows a high precision of functional annotation whereas simple orthology does not.

1.4 The PhyloFacts database: statistics, construction and navigation

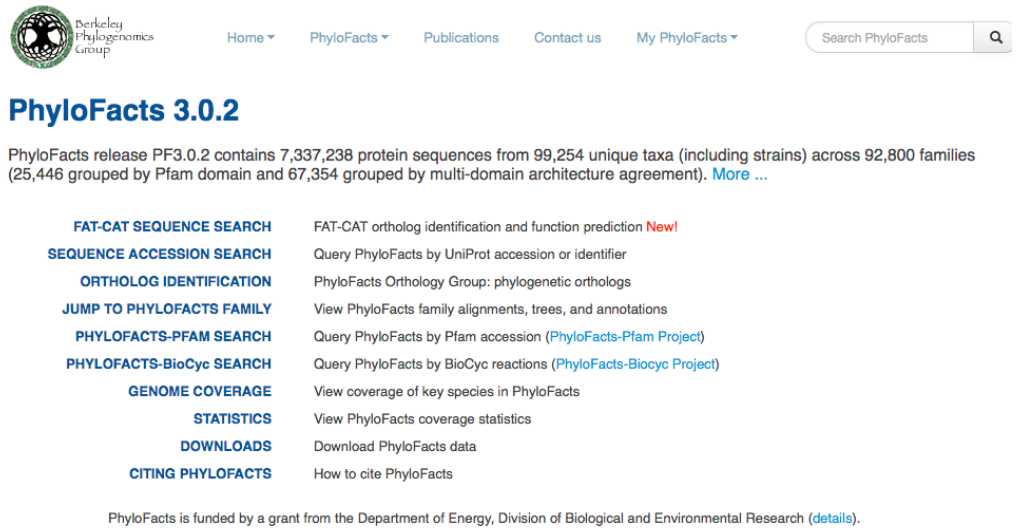


Figure 4. PhyloFacts home page (phylogenomics.berkeley.edu/phylofacts)

The PhyloFacts database has been developed and supported by the Berkeley Phylogenomics Group since 2003, with numerous associated webservers [17, 18, 24].

PhyloFacts 3.0.2 includes trees for virtually all Pfam domains and most multi-domain architectures (MDAs), with >7.3M proteins from 99K distinct taxon IDs (including strains) across from Eukaryotes, Bacteria and Archaea clustered into >93K families. We allow sequences to belong to more than one tree, for two reasons: (1) we create trees for individual Pfam domains as well as for multi-domain architectures; and (2) many protein superfamilies are too large to include all members in a single tree. (The 7TM_1 Pfam family is an example of this type.)

PhyloFacts integrates experimental and annotation data from different resources including SwissProt, the Gene Ontology, Pfam, BioCyc and 3rd-party orthology databases. These data are used to derive a profile of functional descriptions at each subtree node in the PhyloFacts database and to provide functional annotations for user-supplied query sequences to the FAT-CAT web server.

Navigating PhyloFacts:

- **FAT-CAT sequence search**: Input a protein sequence to find orthologs and predict function
- **Sequence accession search**: type in a UniProt or GenBank accession to see if it is in a PhyloFacts family.
- **Jump to PhyloFacts Family**: type in a bpg (Berkeley Phylogenomics Group) accession to go directly to the protein family page
- **PhyloFacts-Pfam search**: enter a Pfam domain name or accession to find PhyloFacts family books for that domain
- **PhyloFacts-BioCyc search**: enter a BioCyc reaction to find PhyloFacts families for that reaction
- **Genome Coverage**: Detailed statistics for selected genomes
- **Statistics**: The number of unique taxa, sequences and other data
- **Downloads**: HMMs, trees and other data

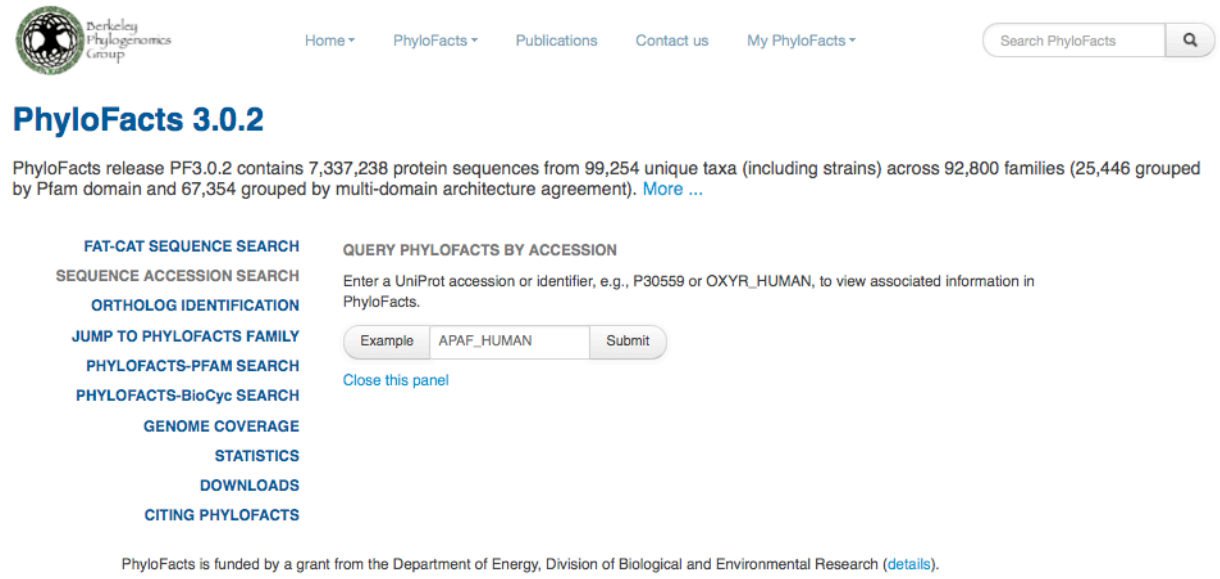


Figure 5. PhyloFacts Sequence Accession input form.

Type in the sequence accession and click Submit.

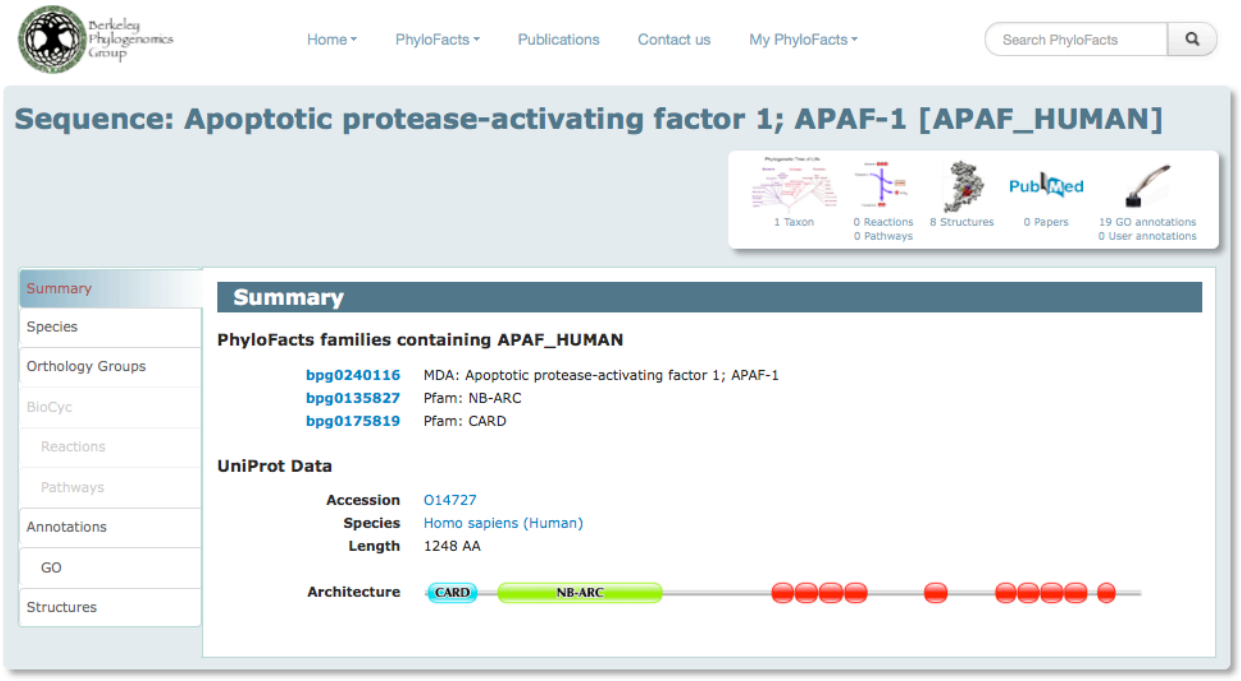


Figure 6. A PhyloFacts sequence page, showing the entry for APAF_HUMAN. APAF_HUMAN is in three PhyloFacts families, one for the multi-domain architecture (MDA) and for the Pfam NB-ARC and Pfam CARD domains.

Statistics and genome coverage. PhyloFacts release PF3.0.2 contains 7,337,238 protein sequences from the UniProt database. Proteins correspond to 99,254 unique taxon identifiers (including strains). Proteins are distributed across 92,800 families: 25,446 families for individual PFAM domains and 67,354 families based on sharing the same multi-domain architecture. PhyloFacts clusters use a set-covering protocol, so that there is some overlap across families (sequences can belong to multiple families). Additional information about representative genomes is available in http://makana.berkeley.edu/phylofacts/coverage/.

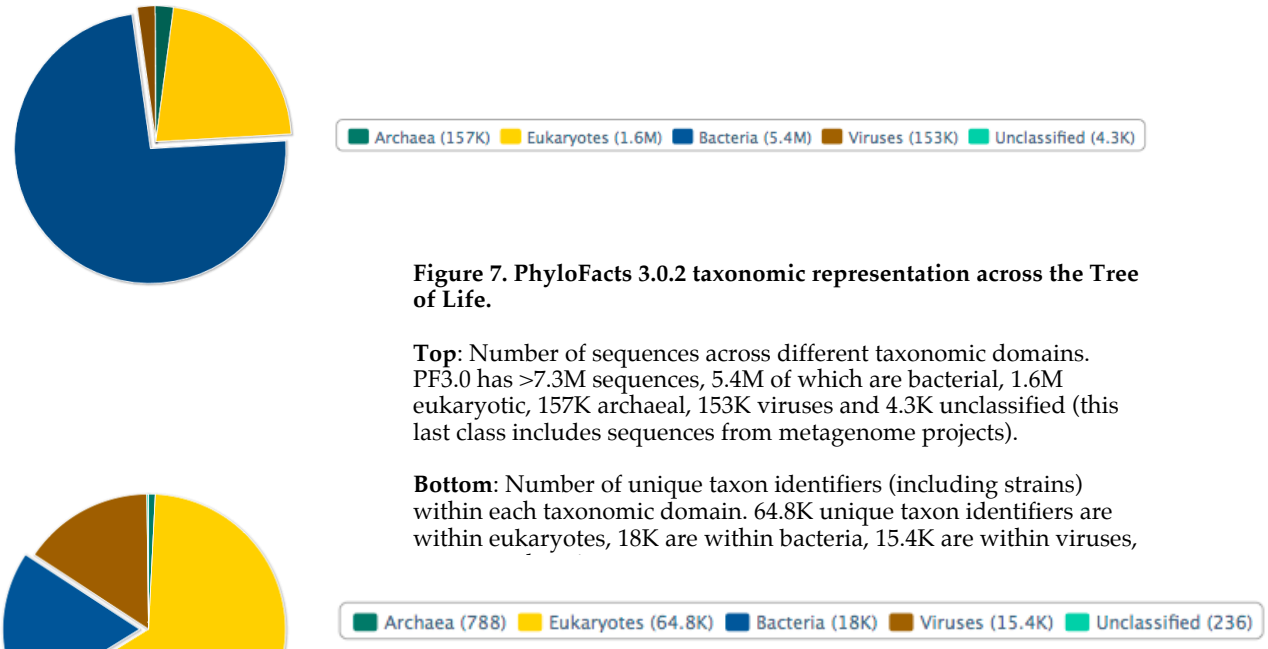


Figure 7. PhyloFacts 3.0.2 taxonomic representation across the Tree of Life.

Top: Number of sequences across different taxonomic domains. PF3.0 has >7.3M sequences, 5.4M of which are bacterial, 1.6M eukaryotic, 157K archaeal, 153K viruses and 4.3K unclassified (this last class includes sequences from metagenome projects).

Bottom: Number of unique taxon identifiers (including strains) within each taxonomic domain. 64.8K unique taxon identifiers are within eukaryotes, 18K are within bacteria, 15.4K are within viruses,

PhyloFacts library construction.

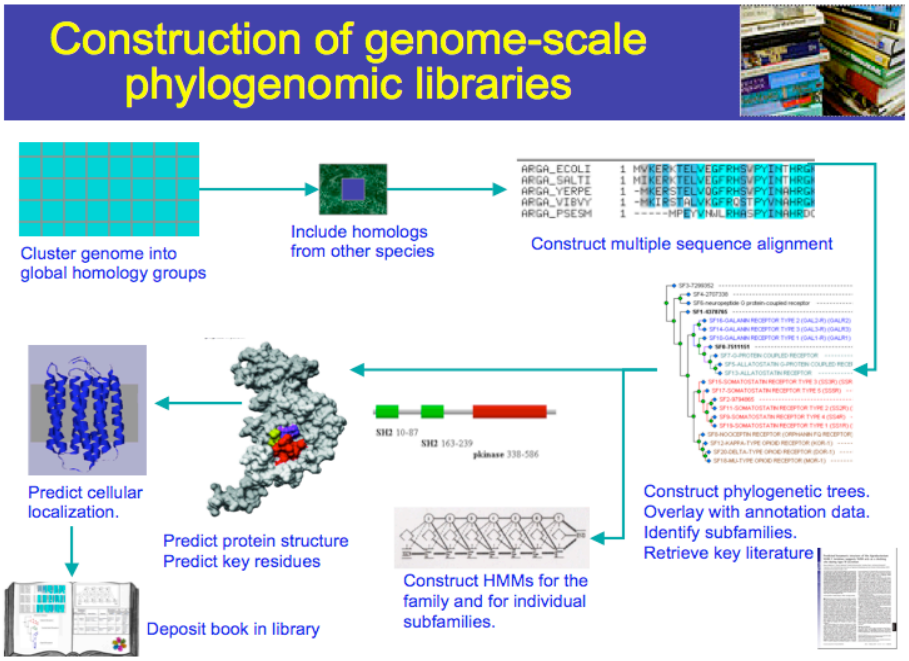


Figure 8. PhyloFacts library construction pipeline.

The PhyloFacts library construction pipeline uses the following protocol to construct a library of books for protein families sharing a common domain architecture.

We first divide a genome into groups based on sharing a common overall domain architecture, followed by FlowerPower [25] to include global homologs from other species. This initial stage is followed multiple sequence alignment construction using MAFFT [26] and phylogenetic tree estimation using FastTree, ortholog identification using PHOG [27] and HMM construction [15]. Gene Ontology annotations and evidence codes are retrieved and overlaid on the tree. Other bioinformatics analyses (using both tools developed by us and 3rd party software) are performed. Details on the pipeline are provided in [18].

The pipeline for constructing domain-based books is identical, except that FlowerPower is parameterized for semi-global clustering.

Two types of PhyloFacts library "book": multi-domain architectures and Pfam domains.

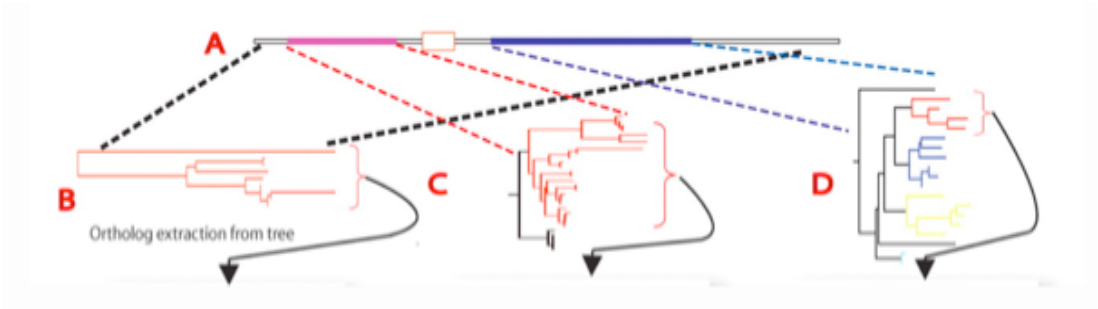


Figure 9. PhyloFacts-Pfam and MDA families: trees for individual domains and whole MDAs. PhyloFacts includes trees for both multi-domain architectures and for individual Pfam domains. Trees for individual Pfam domains are part of the PhyloFacts-Pfam project. In this example, a protein (A) has three domains (pink and blue), and is represented by three trees in PhyloFacts: tree B is for the MDA, while C is for the pink domain and D is for the blue domain.

PhyloFacts-Pfam: PhyloFacts 3.0 includes trees for Pfam domains [28] found in genomes targeted for coverage in our database. We use Pfam-A HMMs to identify matching subregions in proteins encoded in targeted genomes, and use these subregions as seeds to gather homologs from the UniProt resource, construct multiple sequence alignments and estimate phylogenetic trees. In other words, the multiple sequence alignment used to estimate a PhyloFacts-Pfam tree is restricted to a single Pfam domain, and the phylogenetic tree is estimated from that domain only. Because many Pfam domains are found in different domain organizations (multi-domain architectures, or MDAs), it is common for sequences included in a PhyloFacts-Pfam tree to span many different MDAs. Fortunately, sequences that share a common MDA tend to cluster into subtrees, allowing orthology prediction methods to be applied to these domain-based methods. There are some technical challenges since internal nodes of the trees may correspond to gene fusion and fission events, and tree reconciliation programs are not designed to handle these types of data. Nevertheless, domain-based phylogenies provide certain advantages over phylogenies for multi-domain architectures (in which sequences are required to align along their entire lengths); specifically, they can improve ortholog identification recall and precision [29].

PhyloFacts Multi-Domain Architecture (MDA) families: We use the FlowerPower algorithm [25] to retrieve sequences sharing a common multi-domain architecture. FlowerPower is an iterated homology clustering algorithm that uses SCI-PHY [15] to identify subfamilies and subfamily HMMs to select and align new sequences. In each iteration, as new sequences are retrieved and aligned, FlowerPower examines the alignment of candidate family members for agreement with the family consensus structure. The resulting cluster of homologs has both high precision and recall in clustering sequences into multi-domain architecture classes [25].

PhyloFacts library books

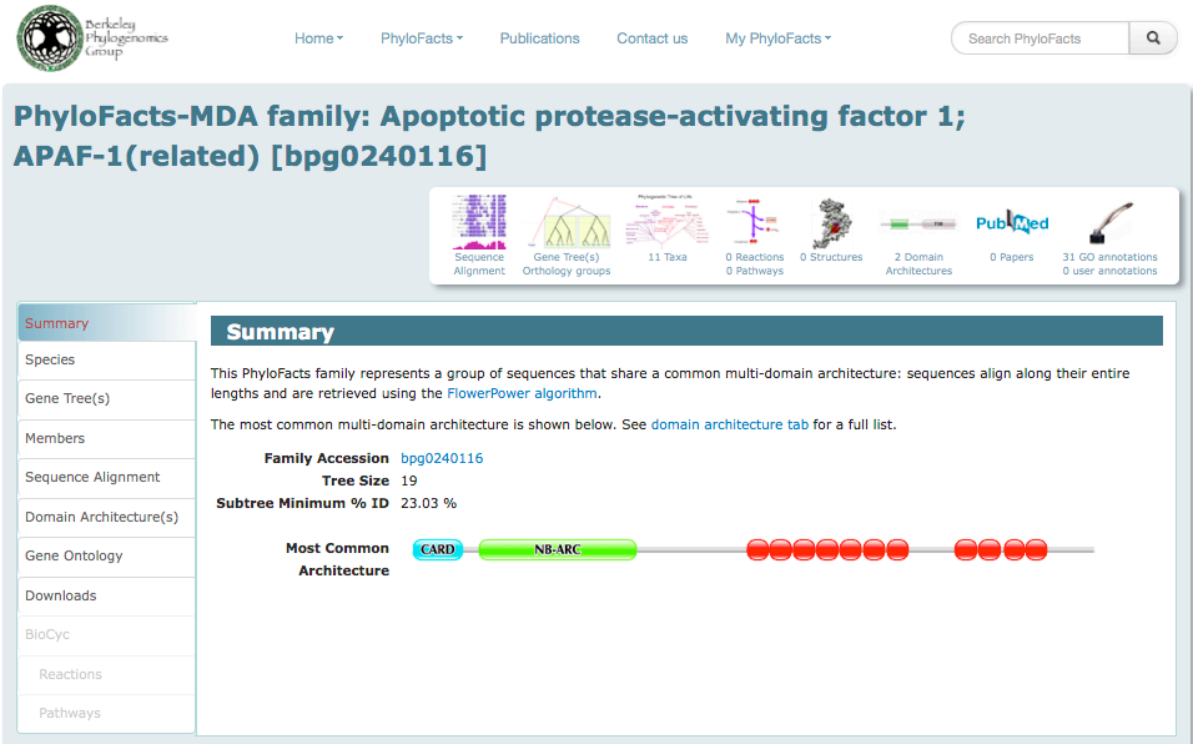
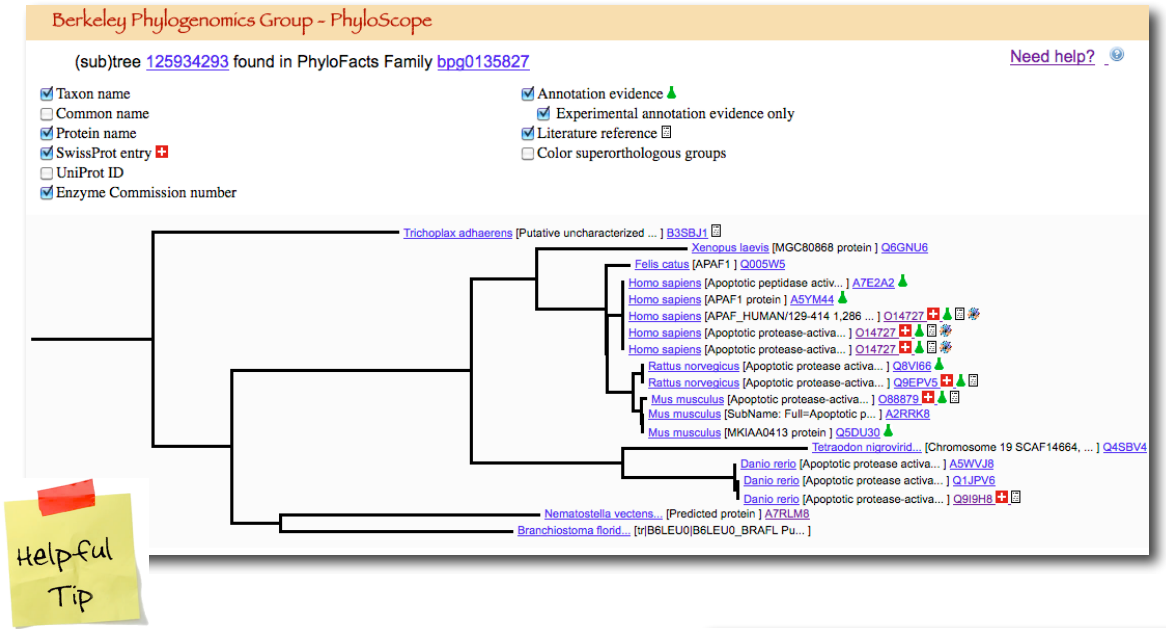


Figure 10. PhyloFacts family book summary view.

Clicking on links in the left navigation bar, or on icons at top, will display additional data for the family.

The PhyloScope Javascript tree viewer

Links to PhyloFacts trees are provided on the Gene tree(s) tab. We provide the Berkeley Phylogenomics Group PhyloScope tree viewer and also the Archaeopteryx viewer provided by Christian Zmasek. PhyloScope is recommended for small to moderate trees (up to ~500 sequences) and Archaeopteryx for larger trees.

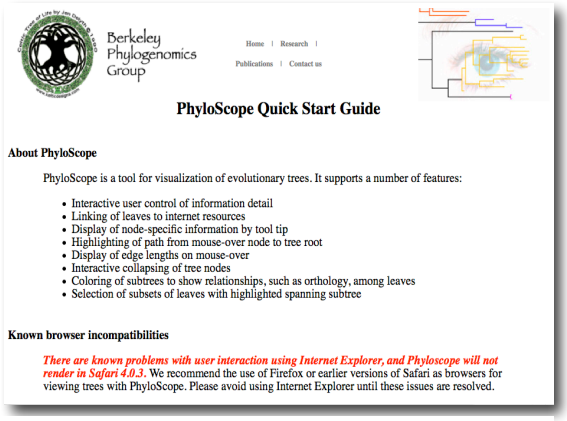


Trees displayed in PhyloScope are decorated with icons indicating experimental support for GO annotations, PDB structures and biological literature; icons are linked to the databases providing these data.

Tool-tip over an icon to see what results are available at that source (see example below, left).

Tool-tip over a sequence or a node in the tree to view the path from the root to that node.

A link to the PhyloScope online help is provided in the upper right corner of the PhyloScope viewer.



PhyloScope help page at
http://makana.berkeley.edu/phyloscope/help/

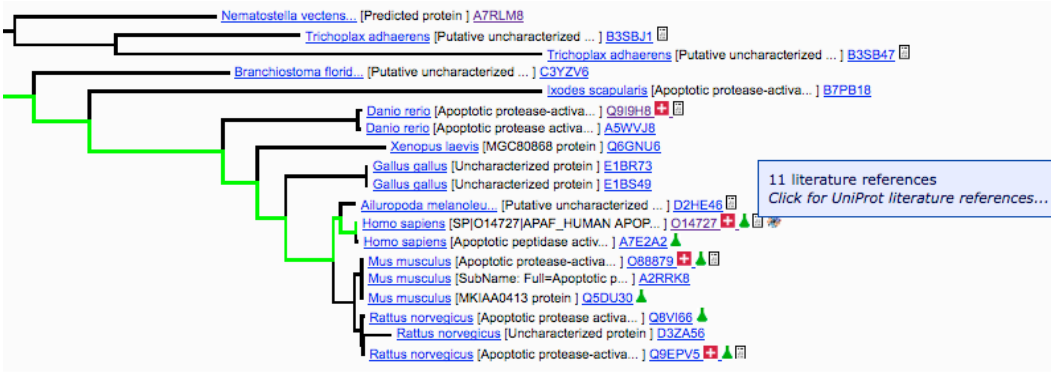


Figure 11. PhyloScope tree viewer

1.5 Subfamily classification using subfamily HMMs

PhyloFacts 2.0 used hidden Markov models at selected subtrees, identified using SCI-PHY [15], to provide functional subclassification of user sequences. We used a logistic regression analysis of the score to the top-scoring subfamily HMM to confirm that the score was in the range observed for subfamily members; if not, the sequence was labeled as a novel subtype.

The HMM parameter estimation protocol used to derive subfamily HMM parameters used an information-tying technique to share statistics across the family; compared to a naïve HMM construction that does not use information across the family, we showed that our information tying protocol boosted recall significantly. However, the naïve approach had a slight improvement over information-tying in precision. Both the information-tying and the naïve protocol made use of Dirichlet mixture densities to include prior information about amino acid substitutions so that HMMs for single sequences or very small families were able to classify novel members.

The FAT-CAT HMM construction uses the HMMER hmmbuild software, and does not use information tying.

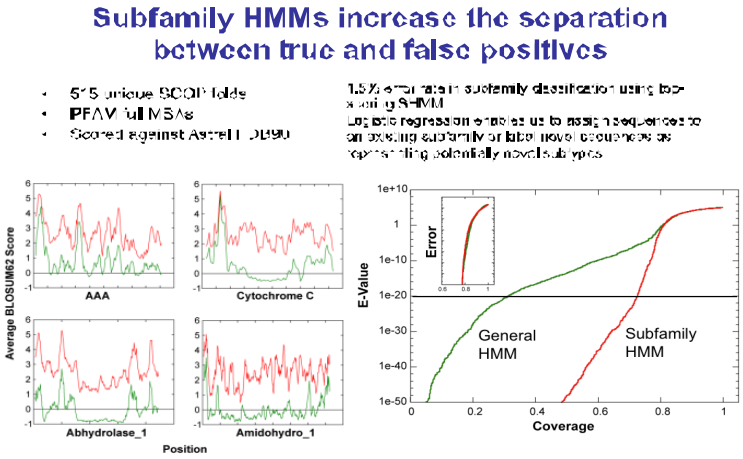
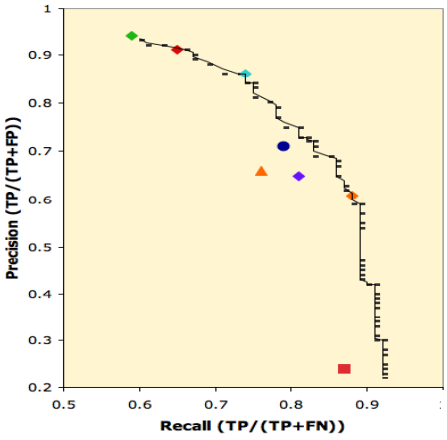


Figure 12. Subfamily HMM performance at remote homolog detection and classification.

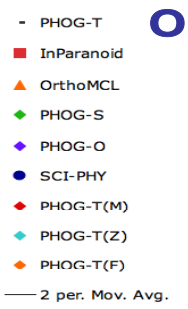
Remote homolog detection benchmarking experiments on the Astral dataset from the Structural Classification of Proteins (SCOP) database. Classification accuracy based on assignment to the top-scoring subfamily HMM has 1.5% error rate, and can be reduced to <1% if logistic regression is used to separate sequences that belong to the family from those that represent novel subtypes. Experiments and details on methodology reported in [15].

1.6 The PHOG (Phylogenomic Orthology Group) algorithm

PHOG analyzes pre-calculated phylogenetic trees to predict orthologs. Results on a manually curated benchmark dataset from the TreeFam resource [30] show that PHOG has superior accuracy to two top-ranked orthology databases (OrthoMCL-DB and Inparanoid). PHOG allows the user to select a desired precision/recall tradeoff by varying a tree distance threshold. Details on the PHOG method and experimental validation are available in [24].



Benchmark dataset 100 (non-homologous) human sequences from TreeFam-A, filtered to remove homologs, and manually curated orthologs from mouse, zebrafish and fruitfly.



Ortholog prediction accuracy

PHOG-O : Standard orthology definition
PHOG-S: Superorthologs
PHOG-T: thresholded PHOGs
PHOG-T(M): optimized for mouse
PHOG-T(Z): optimized for zebrafish
PHOG-T(F): optimized for fruitfly

	Mouse		Zebrafish		Fruit fly		Overall	
Method	Rec	Prec	Rec	Prec	Rec	Prec	Rec	Prec
InParanoid	0.96	0.21	0.81	0.22	0.85	0.29	0.87	0.24
OrthoMCL	0.83	0.77	0.78	0.48	0.68	0.83	0.76	0.66
PHOG-S	0.82	0.94	0.62	0.92	0.38	0.98	0.59	0.94
PHOG-O	0.92	0.58	0.84	0.67	0.70	0.72	0.81	0.65
SCI-PHY	0.96	0.70	0.91	0.66	0.54	0.82	0.79	0.71
PHOG-T(M)	0.95	0.91	0.65	0.86	0.38	0.98	0.65	0.91
PHOG-T(Z)	0.96	0.84	0.84	0.81	0.48	0.98	0.74	0.86
PHOG-T(F)	0.96	0.53	0.91	0.57	0.78	0.78	0.88	0.61

Figure 13. PHOG ortholog identification benchmarked on manually curated orthologs in the TreeFam database

1.7 The FAT-CAT Pipeline

The FAT-CAT pipeline starts with the submission of a protein sequence and parameter selection and proceeds through family and subtree HMM scoring to ortholog identification and functional annotation. The FAST-CAT variant differs from the default FAT-CAT pipeline in stage 3 (indicated by red arrows). In stage 1, the query is scored against family HMMs in the PhyloFacts database for proteins sharing the same multi-domain architecture (MDA, shown at top) and HMMs constructed for Pfam domains (shown at bottom). Families meeting stage 1 criteria (E-value and alignment statistics) are passed to stage 2. In this toy example, PhyloFacts trees for two Pfam domains and a tree for the multi-domain architecture meet stage 1 criteria and are passed to stage 2.

In stage 2, we obtain an approximate phylogenetic placement of the query in each tree by scoring all the HMMs in the tree. The subtree node corresponding to the top-scoring HMM is examined to determine its suitability as a source of orthologs to the query: stage 2 parameters include the query-subtree HMM score and alignment and whether the subtree appears to be restricted to orthologs. For each top-scoring node that meets these criteria we identify a (typically larger) enclosing clade supported by one or more orthology methods. Enclosing clades are passed to Stage 3 for ortholog identification.

In stage 3, FAT-CAT and FAST-CAT diverge. FAT-CAT (blue arrows) evaluates the pairwise alignment between the query and each sequence and identifies all supporting evidence supporting the orthology. FAST-CAT (red arrows) avoids much of this computational complexity by using a fast k-tuple comparison to select the most similar sequences from the enclosing clade, constructing an MSA including the query using MAFFT, estimating a phylogenetic tree using FastTree, and extracting a subtree of the phylogenetically closest sequences (i.e., based on tree distance to the query). Alignment analysis can then be restricted to this smaller subset based on the multiple sequence alignment. Sequences meeting these criteria are passed to stage 4.

In stage 4, we derive a weighted consensus functional annotation for the query based on orthologs selected in stage 3. Annotations from close orthologs are given higher weight than those from more distant orthologs, and manually curated annotations are given higher weight than those that are derived computationally.

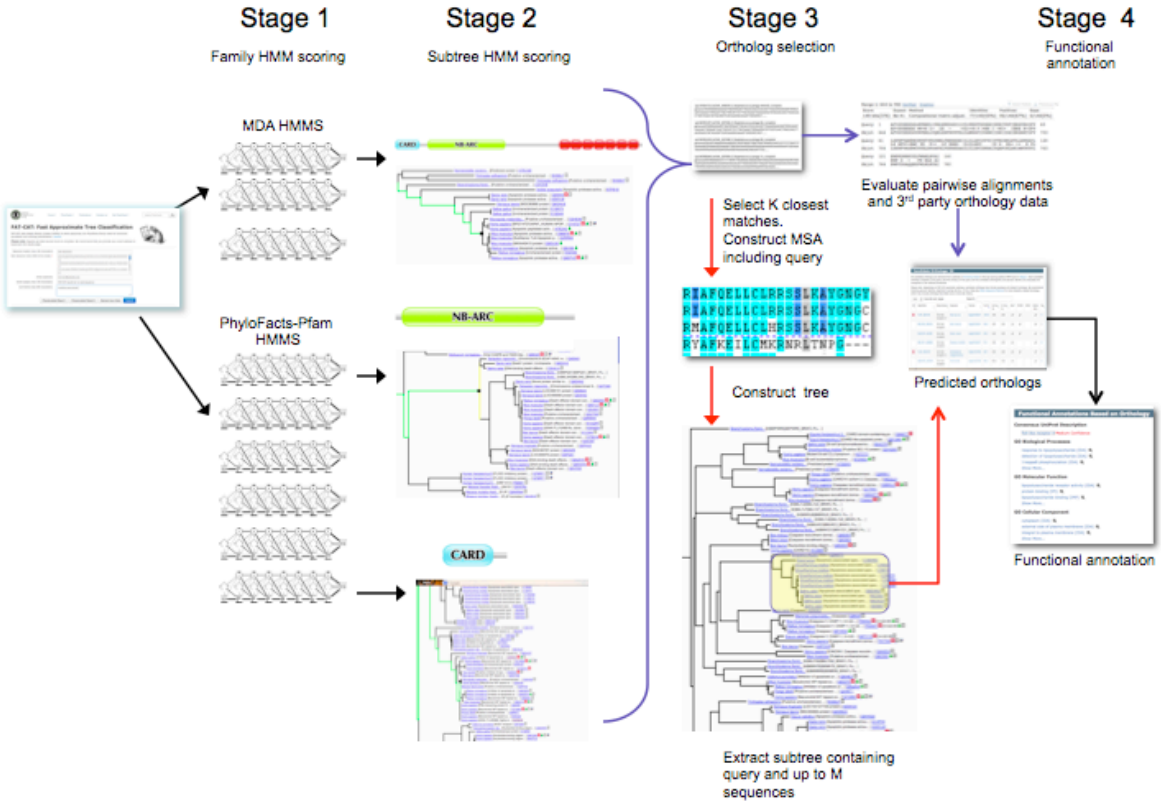


Figure 14. FAT-CAT pipeline. The FAST-CAT variant is shown in red.

2. SUBMITTING A SEQUENCE TO THE FAT-CAT WEBSERVER

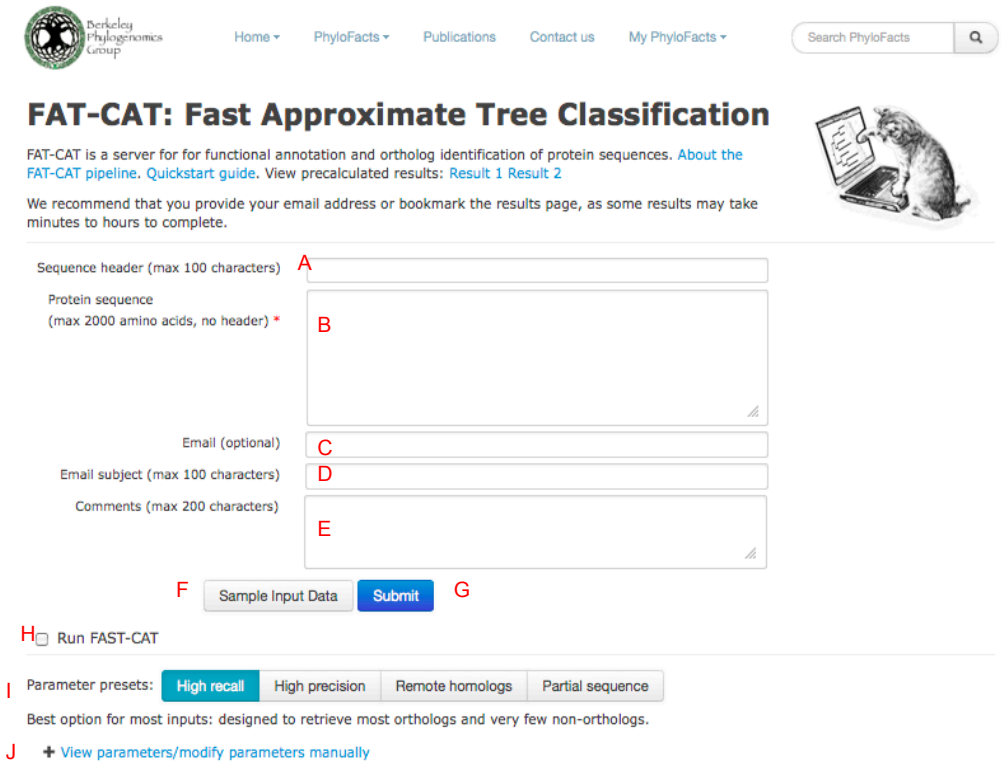


Figure 15. FAT-CAT input form (http://phylogenomics.berkeley.edu/phylofacts/fatcat/).

2.1 The FAT-CAT input form

A: Header line (optional). This is recommended so that you can remember what sequence you submitted.
B: Protein sequence (required). Raw format – no header line, just the amino acids (see Figure 16).
C: Email address (optional). If you'd like a link to results to be sent to you by email, input your email address. Otherwise, bookmark the results page displayed after you click Submit.
D: Email subject (optional): If you leave this empty, the subject line will read "PhyloFacts FAT-CAT Job <number> has completed".
E: Comments (optional). Comments are stored in the Job Summary section of the Results.
F: Sample Input Data. Clicking on this button populates the form so that you can see what kind of input is expected in each section.
G: Submit. Click on this when you're ready to launch the job. It will bring you to a progress page where you can track the progress of your job.
H: FAST-CAT. FAST-CAT is a beta version of FAT-CAT designed for speed.
I: Parameter presets. We provide four different parameter settings for different types of input sequences. Click on each button to see a brief description of the types of inputs these settings are designed to handle.
J: View/Modify parameters. Click on this link to view and edit any individual parameters.

2.2 FASTA and raw sequence formats

Most bioinformatics web servers allow either FASTA input (in which the first line is a "header line" starting with a ">" symbol) or raw sequence, which omits the header line. The FAT-CAT webserver expects raw sequence – no header line – in the input box. You can put the header line into the corresponding section of the input form (optional).

```
Example FASTA input:

>sp|Q9XT58|ADRB3_SHEEP Beta-3 adrenergic receptor OS=Ovis aries GN=ADRB3 PE=3 SV=2
MAPWPPGNSSLTPWPDIPTLAPNTANASGLPGVPWAVALAGALLALAVLATVGGNLLVIV
AIARTPRLQTMTNVFVTSLATADLVVGLLVVPPGATLALTGHWPLGVTGCELWTSVDVLC
VTASIETLCALAVDRYLAVTNPLRYGALVTKRRARAAVVLVWVVSAAVSFAPIMSKWWRV
GADAEAQRCHSNPRCCTFASNMPYALLSSSVSFYLPLLVMLFVYARVFVVATRQLRLLRR
ELGRFPPEESPPAPSRSGSPGPAGPYASPAGVPSYGRRPARLLPLREHRALRTLGLIMGT
FTLCWLPFFVVNVVRALGGPSLVSGPTFLALNWLGYANSAFNPLIYCRSPDFRSAFRRLL
CRCPPEEHLAAASPPRAPSGAPTVLTSPAGPRQPSPLDGASCGLS

The raw sequence for this entry is:

MAPWPPGNSSLTPWPDIPTLAPNTANASGLPGVPWAVALAGALLALAVLATVGGNLLVIV
AIARTPRLQTMTNVFVTSLATADLVVGLLVVPPGATLALTGHWPLGVTGCELWTSVDVLC
VTASIETLCALAVDRYLAVTNPLRYGALVTKRRARAAVVLVWVVSAAVSFAPIMSKWWRV
GADAEAQRCHSNPRCCTFASNMPYALLSSSVSFYLPLLVMLFVYARVFVVATRQLRLLRR
ELGRFPPEESPPAPSRSGSPGPAGPYASPAGVPSYGRRPARLLPLREHRALRTLGLIMGT
FTLCWLPFFVVNVVRALGGPSLVSGPTFLALNWLGYANSAFNPLIYCRSPDFRSAFRRLL
CRCPPEEHLAAASPPRAPSGAPTVLTSPAGPRQPSPLDGASCGLS

The header line is: >sp|Q9XT58|ADRB3_SHEEP Beta-3 adrenergic receptor OS=Ovis aries
GN=ADRB3 PE=3
```

Figure 16. FASTA and raw sequence formats. FAST-CAT requires raw sequence input (amino acid only).

2.3 What types of sequences does FAT-CAT accept?
Amino acid sequences only. The maximum sequence length is 2000 amino acids.

2.4 Submitting your email address or bookmarking the results page.
We encourage you to submit your email address, as some FAT-CAT jobs can take an hour or longer to return. If you don't want to give your email address, then bookmark the page that displays after you click Submit. Or copy the URL and save it for your records.

2.5 Tracking the progress of your job.
After you click Submit, you will be brought to a page where you can track the progress of your submitted job. You will be reminded to bookmark that page if you haven't submitted your email address. As the job progresses through the four stages the text will change.

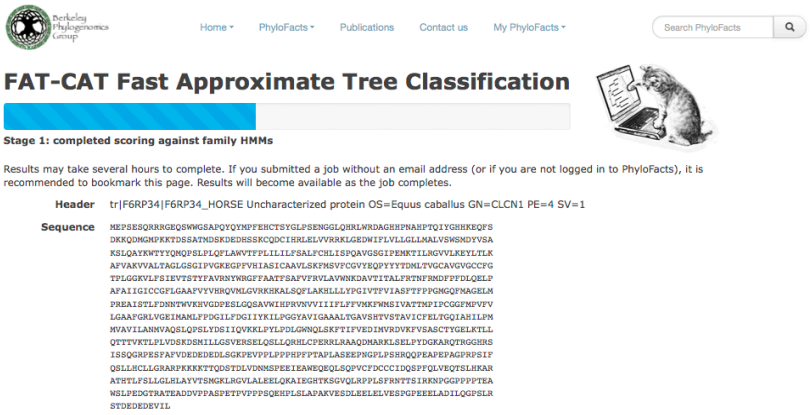


Figure 17. The FAT-CAT progress page. This page displays after you click Submit. You can track your job progress on this page. If you haven't provided your email address, you should bookmark this page.

3. SELECTING AND MODIFYING FAT-CAT PIPELINE PARAMETERS

Pipeline parameters for stages 1 through 3 are designed to accommodate different types of input sequences.

- **High Recall** parameters are designed to handle sequences that are full-length, contain no promiscuous domains and have no close paralogs (e.g., with high sequence identities).
- **High Precision** parameters are designed for cases where a sequence is known to contain a promiscuous domain (e.g., a kinase domain, leucine-rich repeat, or other commonly found structural domain) or sequences with close paralogs (e.g., with >70% identity).
- **Partial Sequence** parameters are designed to handle fragments, partial sequences and splice variants.
- **Remote Homolog Detection** parameters handle cases where few or no homologs, or only matches that align over local regions, can be found by BLAST. In this last case, however, we remind users that results will represent distant homologs that may not have the same function and may not be actual orthologs.

3.1 A decision tree to guide parameter selection.

The decision tree shown in Figure 18 is included to help you choose among pre-set parameter settings.

3.2 What if you don't know anything about your query?

The decision tree shown in Figure 18 assumes you know something about your query. But what if you have no idea how to answer those questions? You have two options. You can just go ahead and submit your sequence to FAT-CAT and see what happens. Or, you can do a little bioinformatics analysis to try to get those answers. BLAST and Pfam will be the primary tools.

Is your sequence complete? Gene model errors and splice variants are quite common, especially in eukaryotes. If your sequence is annotated as a partial sequence or fragment, it's not complete. But many sequences that are annotated as complete are actually incomplete. To figure out which is which, try running BLAST and examine close matches. If BLAST results include close homologs that are roughly the same length as your query and align well along their entire length, your sequence is probably complete. But: if close homologs are much longer than your query, your sequence may be partial.

Does your sequence have a promiscuous domain? Promiscuous domains are defined by their propensity to be found in many different multi-domain architectures or arrangements; this makes them a source of functional annotation error based on annotation transfer, and also causes problems with orthology prediction. Again, if BLAST matches include many sequences that align along one region of your query, or are much longer or shorter than your query, your query may contain a promiscuous domain. Try submitting your query to Pfam (e.g., at http://pfam.sanger.ac.uk/) to see what Pfam domains are present. For each Pfam domain found, check the Domain Arrangements tab – if any Pfam domains in your query are found in many different domain arrangements, your query has a promiscuous domain.

Does your sequence have close paralogs? Duplication events are quite common, and if they occur fairly recently in evolutionary history so that paralogs have high sequence identity (e.g., >70% identity), it can be easy to confuse paralogs with orthologs. If BLAST analysis lists many proteins with obviously different functions, your query may be in a superfamily with recent duplication events. Examples of this type include different ion channels, G-protein coupled receptors, Toll-like receptors, globins, and defensins.

Tip: You can submit your sequence to FAT-CAT using the default settings, set for high recall. If too many results are returned or paralogs are included in the candidate orthologs, resubmit using the High Precision settings. If too few results are returned or you see orthologs are included in the "Other Sequence Matches" tab, modify parameters manually to relax sequence identity or alignment overlap requirements. You can also try the Remote Homolog Detection settings.

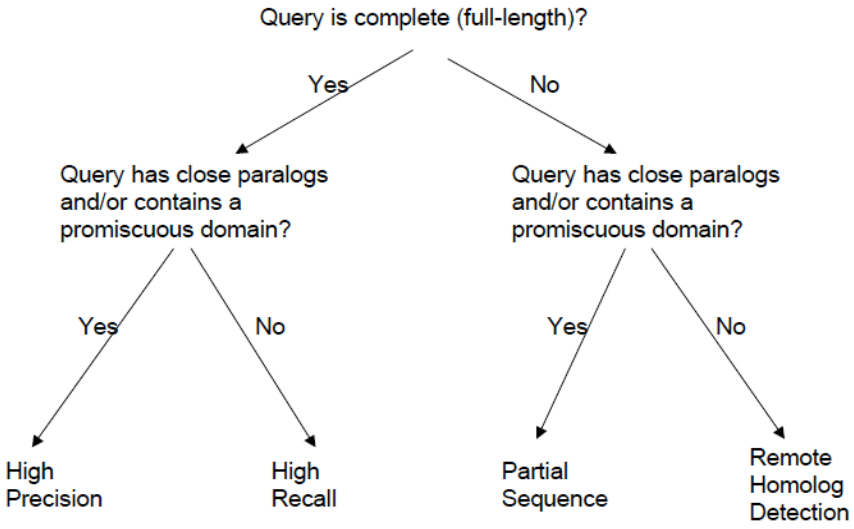


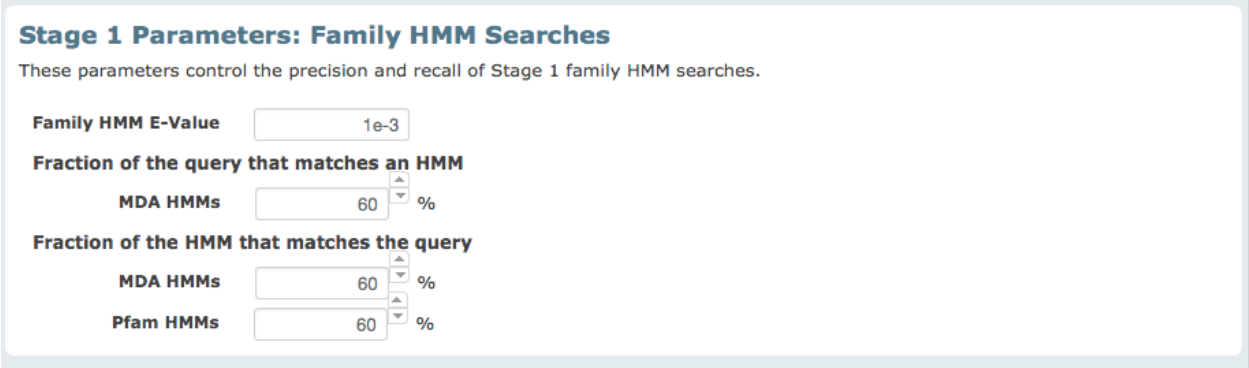
Figure 18. FAT-CAT decision tree for selecting pipeline parameter presets.
Pipeline parameters for stages 1 through 3 are designed to accommodate different types of input sequences. Sequences that are full-length, contain no promiscuous domains and have no close paralogs (e.g., with high sequence identities) are the easiest; the default parameter settings of High Recall will give excellent results for these inputs. If a sequence is known to contain a promiscuous domain (e.g., a kinase domain, leucine-rich repeat, or other commonly found structural domain) then the High Precision settings will produce fewer errors. Sequences that are partial subsequences of full-length proteins or which correspond to splice variants require special handling. The Partial Sequence settings are designed to handle sequences having detectable homologs, while the Remote Homolog Detection settings handle cases where few or no homologs can be found by BLAST. In this last case, however, we remind users that results will represent distant homologs that may not have the same function and may not be actual orthologs.

3.3 Understanding and modifying pipeline parameters.

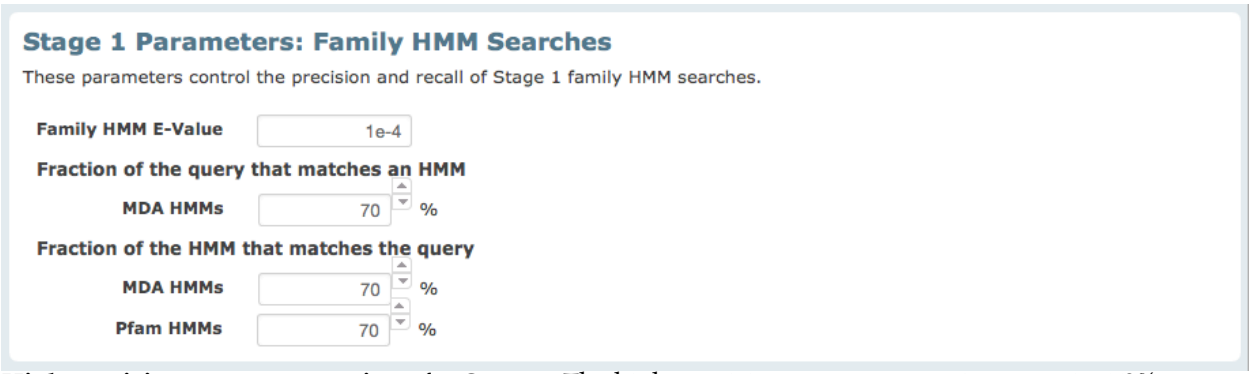
Stage 1 parameters evaluate the family HMM scores and possibly also the alignment between the query and the family HMM. Those that pass stage 1 criteria are sent to stage 2. In stage 2, we score the query against all the HMMs in each tree that pass stage 1. HMMs are placed at every node in the tree, including the leaves (corresponding to single sequences). Stage 2 parameters evaluate the subtree HMM scores and possibly also the alignment statistics. In Stage s we evaluate the characteristics of the subtree whose HMM gave the query the strongest score and also the alignment between the query and the subtree HMM. Those subtrees meeting these criteria are used to identify enclosing clades containing additional sequences that may be orthologous to the query. Details on how we define enclosing clades is found in section 2.4. Enclosing clade sequences are retrieved and analyzed in stage 3 and separated into candidate orthologs and "other sequence matches" based on meeting or failing these criteria.

3.3.1 Stage 1 parameters

Stage 1 parameters control which families are examined in Stage 2 – the more stringent the E-value and coverage criteria, the fewer families are examined. More stringent parameters reduce the run-time, but may also reduce recall.

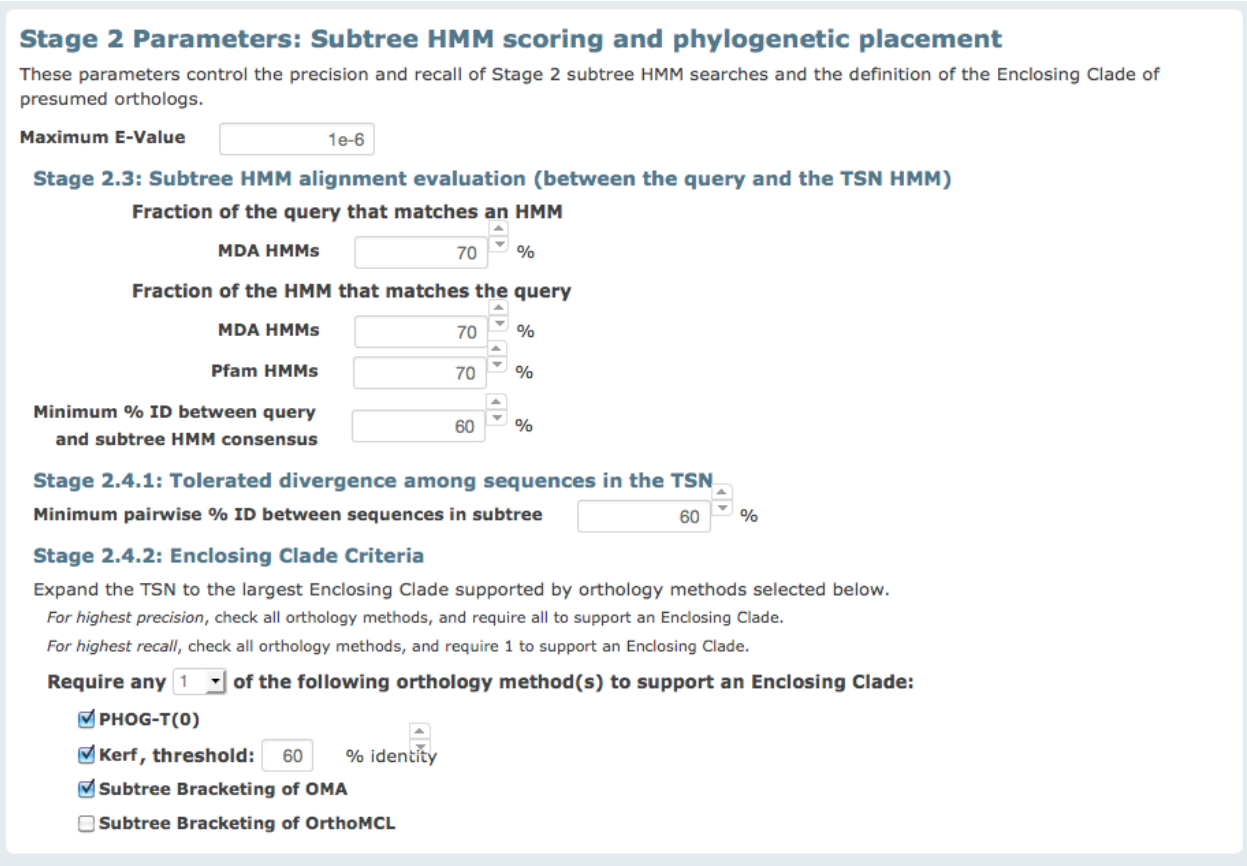


High recall parameter settings for Stage 1. High Recall settings require only a significant E-value and do not enforce minimum coverage of the HMM or query.



High precision parameter settings for Stage 1. The high-precision parameter settings require 70% coverage of Pfam HMMs and 70% bi-directional coverage of the query and MDA HMM. MDA bi-directional coverage requirements enable us to assume that the query and MDA family agree along their entire lengths and share the same multi-domain architecture.

3.3.2 Stage 2 parameters



Stage 2 parameters for high precision. In stage 2, the query is scored against subtree HMMs and the top-scoring HMM is identified. The HMM E-value is evaluated and the alignment between the query and subtree HMM is examined for agreement with defined fractional overlaps and percent identity. The top-scoring node (TSN) is evaluated to determine whether it is found in a clade of possible orthologs. Parameter settings in this stage affect which top-scoring nodes are accepted as a source of possible orthologs in stage 3.

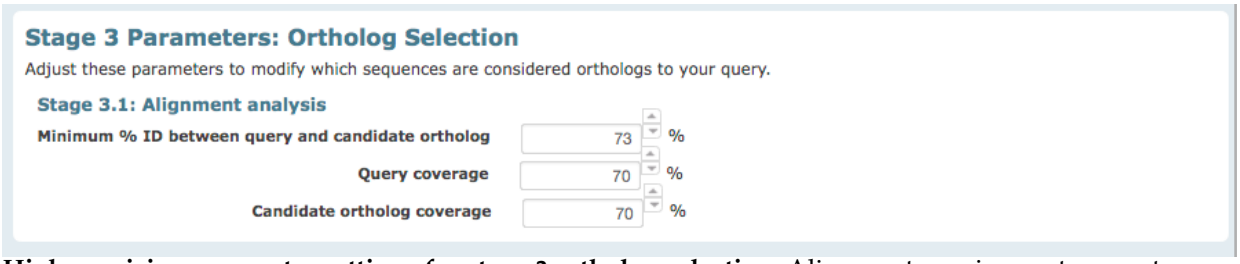
Stage 2.4.2: Enclosing clade criteria. We find the largest enclosing clade supported by the orthology method selected. Requiring only one of the orthology methods to support an enclosing clade produces the largest number of candidate orthologs to be forwarded to stage 3 for analysis, and increases the time required for processing. Requiring more than one method increases stringency, improving precision but lowering recall. See Figure 19.

We use two methods developed by the Berkeley Phylogenomics Group – PHOG and Kerf. PHOG (Figure 13) and Kerf divide PhyloFacts trees into subtrees using phylogenetic tree distances, sequence divergence within subtrees and taxonomic origin information. Subtree bracketing overlays third-party orthology database data on PhyloFacts trees to find subtrees that are supported by those data (subtree bracketing is described on the next page and illustrated in Figure 19).

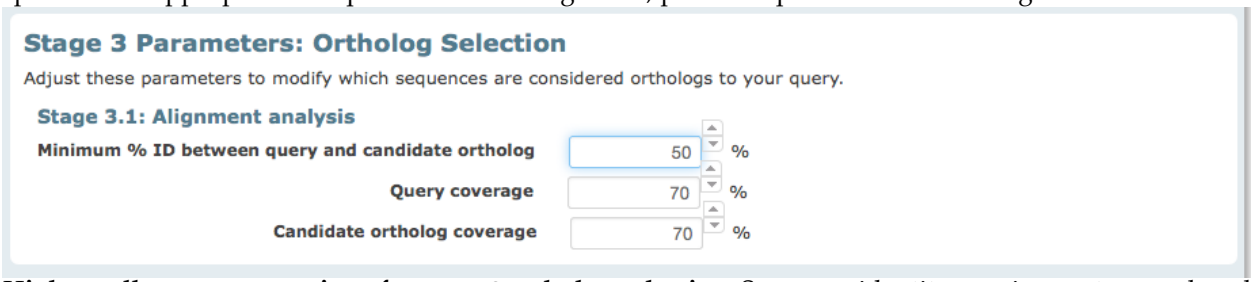
You can control the precision and recall of the enclosing clade by modifying the Kerf threshold. To increase the size of the enclosing clade, reduce the sequence identity cutoff for Kerf. Kerf cuts a tree into subtrees based on the observed sequence divergence within each subtree. A Kerf cutoff of 60% identity will find a subtree including the top-scoring node where no pair of sequences has <60% identity. If you set the Kerf cutoff to 50%, the enclosing clade will be larger.

3.3.3 Stage 3 parameters for ortholog selection

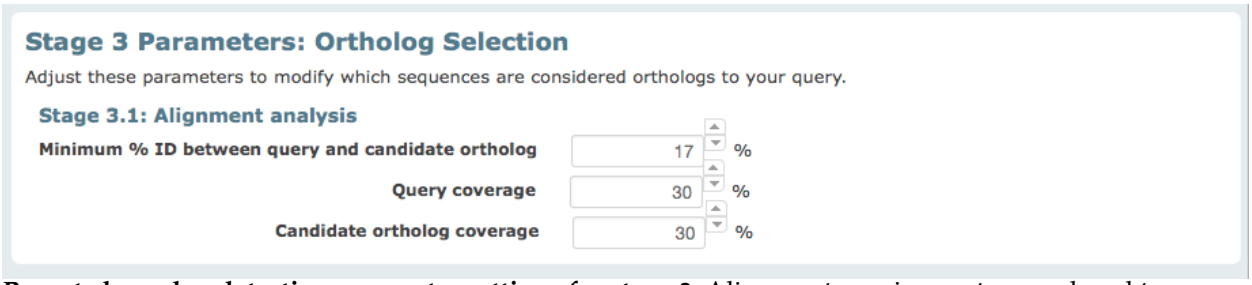
Parameter settings in stage 3 evaluate the alignment of the query and each sequence retrieved from one or more enclosing clades found in stage 2. Sequences meeting the requirements are displayed in "Candidate Orthologs" and those failing one or more criteria are displayed in "Other Sequence Matches".
Important note: If multiple sequences from the same genome are found in the enclosing clades, we select the sequence(s) with the highest sequence identity to the query as the presumed ortholog.



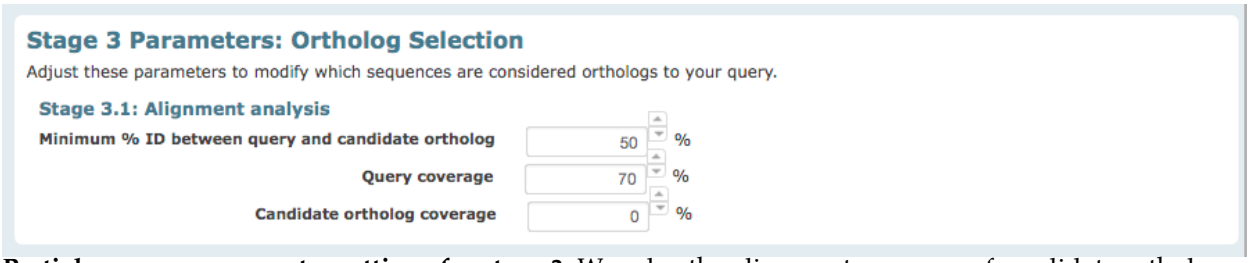
High-precision parameter settings for stage 3 ortholog selection. Alignment requirements are set stringently. These prevent the inclusion of closely related paralogs but can exclude orthologs from distant species. Not appropriate for queries that are fragments, partial sequences or that have gene model errors.



High-recall parameter settings for stage 3 ortholog selection. Sequence identity requirements are relaxed to include orthologs from taxonomically distant species.



Remote homolog detection parameter settings for stage 3. Alignment requirements are relaxed to identify distant homologs with partial (local) alignments. Matches are likely to include paralogs and may have different multi-domain architectures.



Partial sequence parameter settings for stage 3. We relax the alignment coverage of candidate orthologs under the assumption that the query may represent a partial subsequences of an actual full-length protein.

3.4 Defining the enclosing clade for a top-scoring subtree node

When the query sequence is closely related (or is an exact match) to a sequence in a PhyloFacts tree, the top-scoring HMM is often located at or near a leaf in the tree. In these cases, we want to retrieve sequences that are potentially outside the subtree defined by that top-scoring node but are within a clade that is restricted to orthologs. To find that enclosing clade, we use various analyses, making use of two methods developed by the Berkeley Phylogenomics Group (PHOG and Kerf) and also integrating orthology predictions from 3rd-party orthology databases. PHOG and Kerf operate directly on phylogenetic trees in PhyloFacts, finding subtrees that meet specific criteria, but integrating 3rd-party orthology data requires a unique approach we call subtree bracketing. Subtree bracketing is illustrated in the figure below.

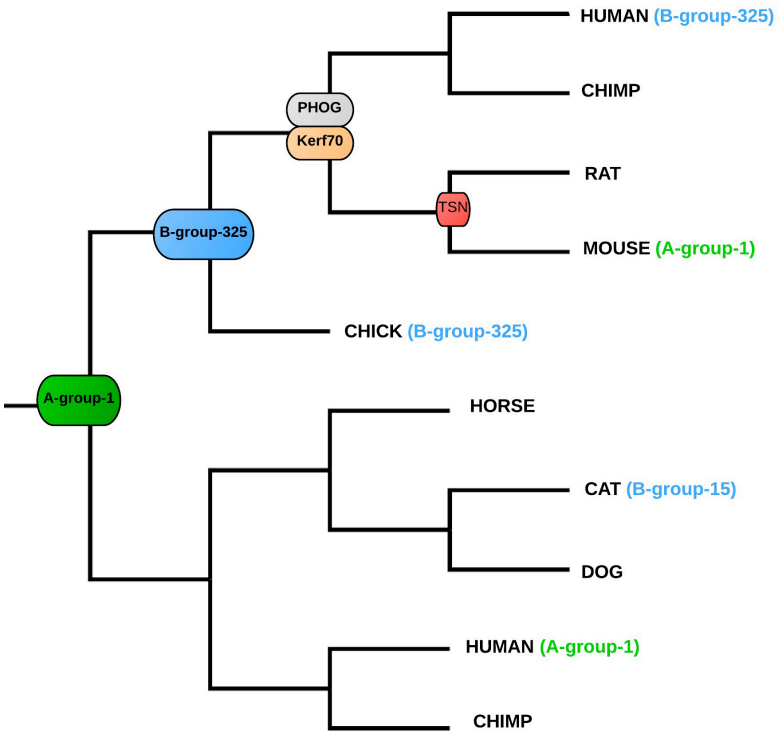


Figure 19. Subtree bracketing and enclosing clades definitions.

This toy example illustrates how we include third-party orthology data in defining an enclosing clade, along with two methods from the Berkeley Phylogenomics Group (PHOG and Kerf).

The top-scoring node (TSN) is shown in red, at the subtree joining rat and mouse sequences. The TSN is just below a node supported by PHOG (using a tree-distance threshold of zero) and Kerf (using a 70% minimum allowed sequence divergence); both PHOG-T(0) and Kerf(70) are conservative protocols, producing very restrictive cuts of trees into subtrees. In this toy example, PHOG and Kerf miss the chicken ortholog.

Two third-party orthology database methods have also been included and overlaid on this tree. Method B (in blue) identifies two distinct orthology groups, B-group 15 and B-group 325. Method A (green) classifies sequences across the tree as belonging to the same orthology group, A-group-1. The top-scoring node is within both the B-group-325 and A-group-1.

The precision and recall of the enclosing clade can be controlled by requiring one or more orthology methods to support the enclosing clade. The least restrictive approach picks the largest enclosing clade for the TSN that is supported by any orthology method. In this toy example, this would result in the entire tree being selected as the enclosing clade (resulting in clear errors – since there are human and chimpanzee sequences in both subtrees). If we require any two methods to agree, the enclosing clade would be rooted at the node labeled B-group-325, which is supported by both A-group-1 and B-group-325. If we require any three methods to agree, the enclosing clade would be rooted at the node labeled by both PHOG and Kerf70.

4. INTERPRETING FAT-CAT RESULTS: A CASE STUDY.

The figure below presents FAT-CAT results for gi|344266516|ref|XP_003405326.1, a predicted apoptotic protease-activating factor 1 isoform 1 from *Loxodonta africana* (African elephant). This result corresponds to PhyloFacts job 2570, and is available at http://makana.berkeley.edu/phylofacts/fatcat/2570/.

In this example, we used the FAT-CAT program defaults (as of 2/8/13), designed for high recall.

4.1 Summary of results

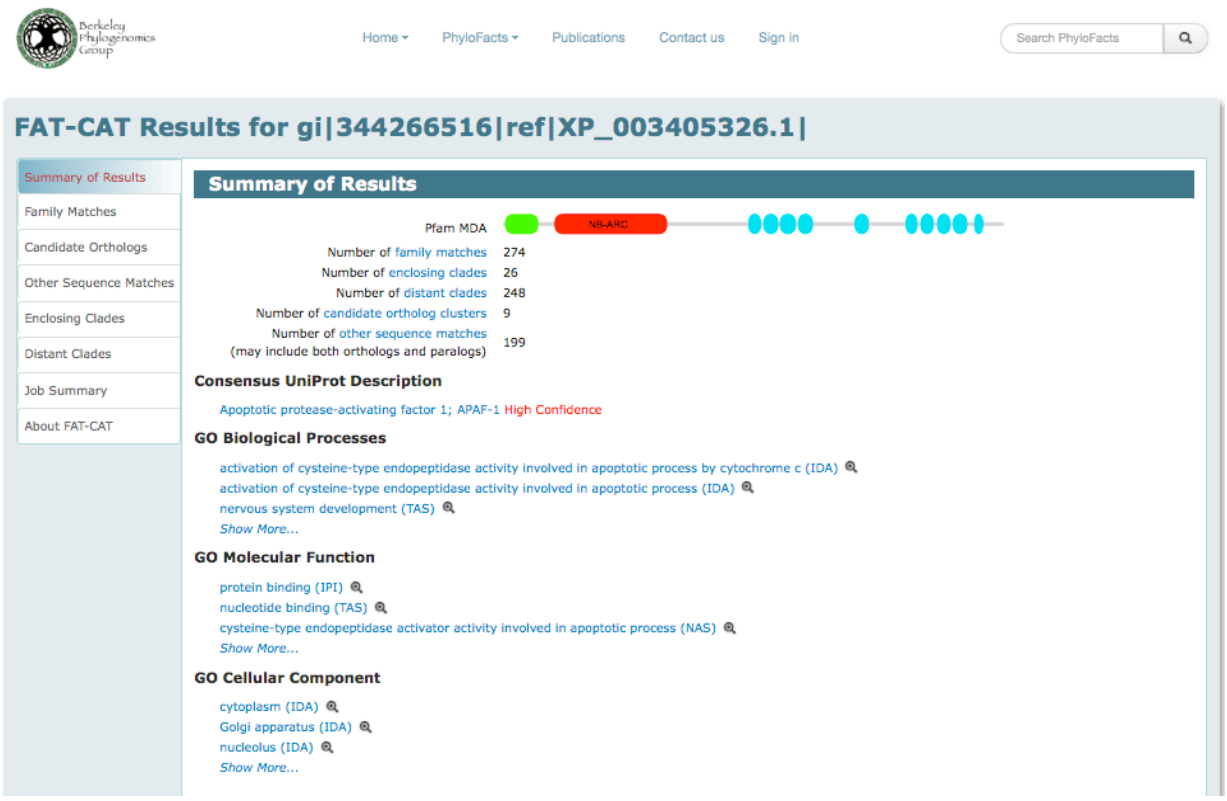


Figure 20. FAT-CAT results page, displaying the Summary of Results: overview of results, including the Pfam multi-domain architecture for the query produced by scanning Pfam-A HMMs. The FAT-CAT pipeline identified 274 families matching Stage 1 criteria, orthologs from nine different genomes (candidate ortholog clusters), and 199 additional homologs that failed one or more criteria for orthology. Predicted functional annotations for the query derived from orthologs satisfying stage 3 criteria are displayed. The Job Summary tab displays the input sequence and all pipeline parameters.

4.2 Enclosing clades

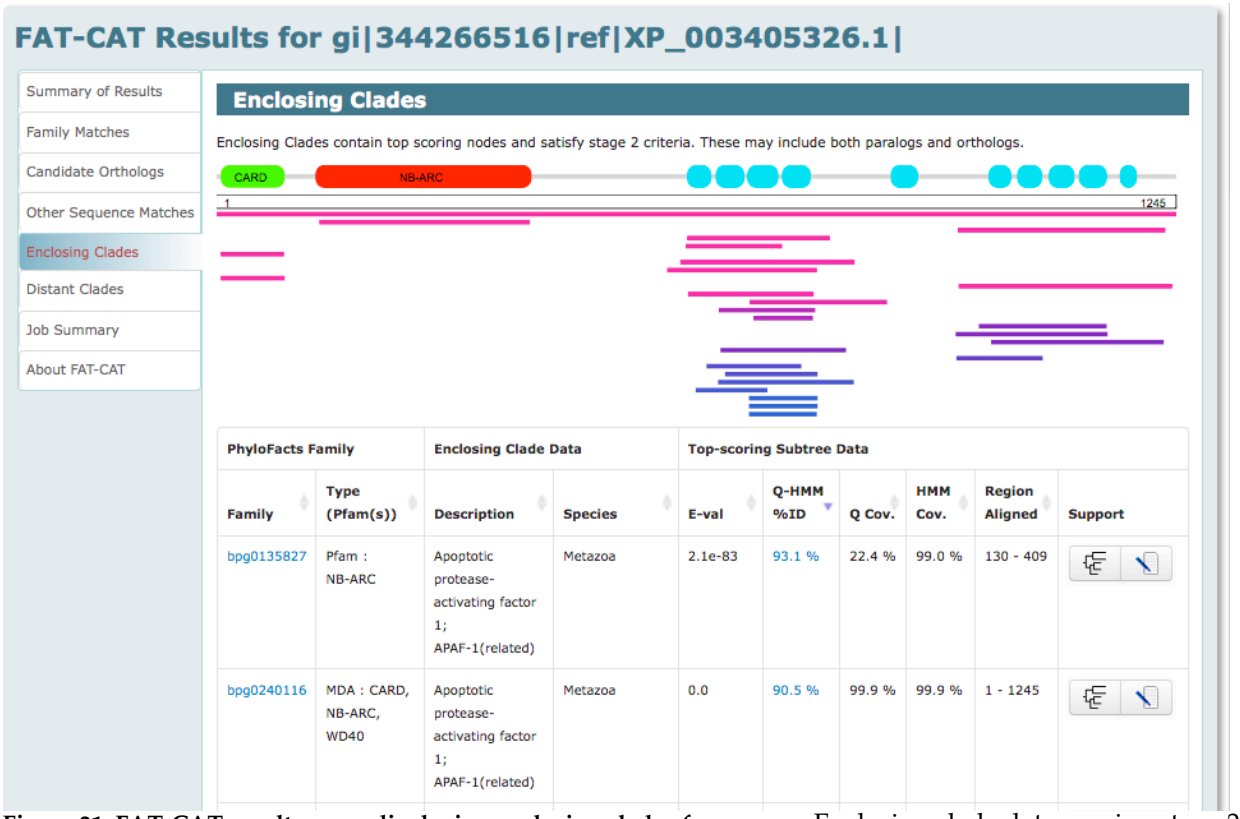


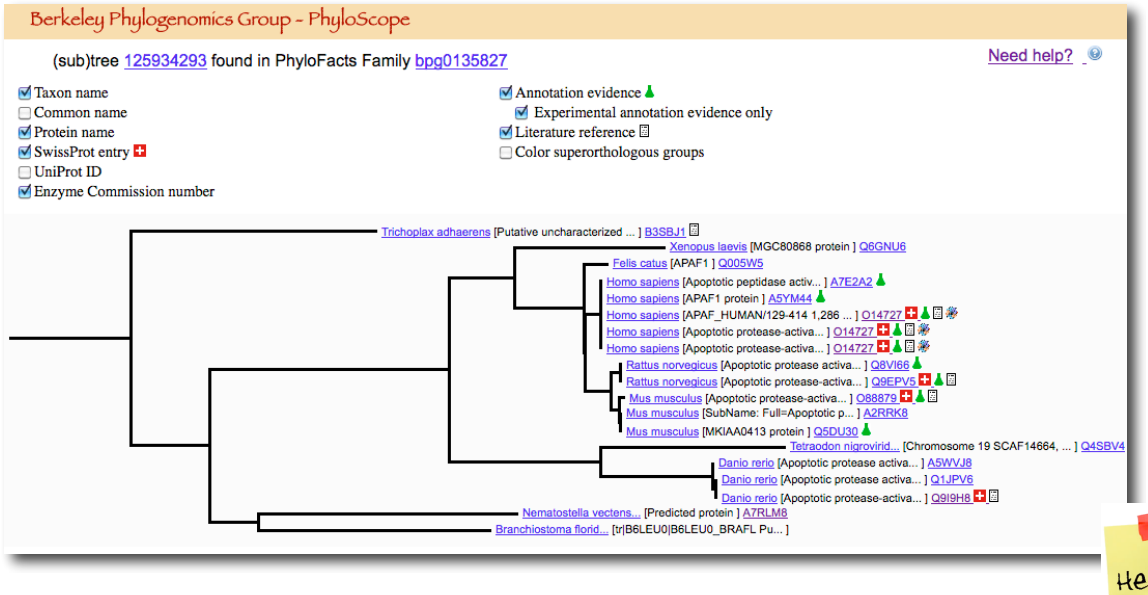
Figure 21. FAT-CAT results page, displaying enclosing clades for a query: Enclosing clade data passing stage 2 criteria, displaying the top two matches sorted by the sequence identity between the query and the top-scoring subtree HMM. The top-scoring HMM matches the PhyloFacts-Pfam NB-ARC domain HMM. The second top-ranked HMM matches along the entire multi-domain architecture (MDA). Clicking on the Q-HMM %ID will display the alignment between the query and HMM. The Family column at far left providesa link to the PhyloFacts family containing that subtree (and lists the bpg accession), and the two icons at far right (tree and page icons) are linked to a phylogenetic tree viewer for the enclosing clade and a link to the PhyloFacts page for the subtree corresponding to the top-scoring node.

Viewing the tree for an enclosing clade. Clicking on the tree icon at the far right of the enclosing clade data table launches the PhyloScope viewer, displaying the enclosing clade and highlighting a path to the top-scoring node (based on stage 2 HMM scoring). In many cases this phylogenetic placement provides an approximate taxonomic classification as well as a functional classification, as shown in this case study.

Tree icons on FAT-CAT pages are linked to the PhyloScope tree viewer. Click on these icons to view trees for enclosing clades.

4.3 View enclosing clades in PhyloScope

Links to PhyloFacts trees for enclosing clades are provided on the Enclosing Clades tab, indicated by tree icons in the far right of the data table. Corresponding tree icons are displayed on the Distant Clades table for top-scoring subtree nodes.



Trees displayed in PhyloScope are decorated with icons indicating experimental support for GO annotations, PDB structures and biological literature; icons are linked to the databases providing these data. A link to the PhyloScope online help is provided in the upper right corner of the PhyloScope viewer.

4.4 Candidate orthologs

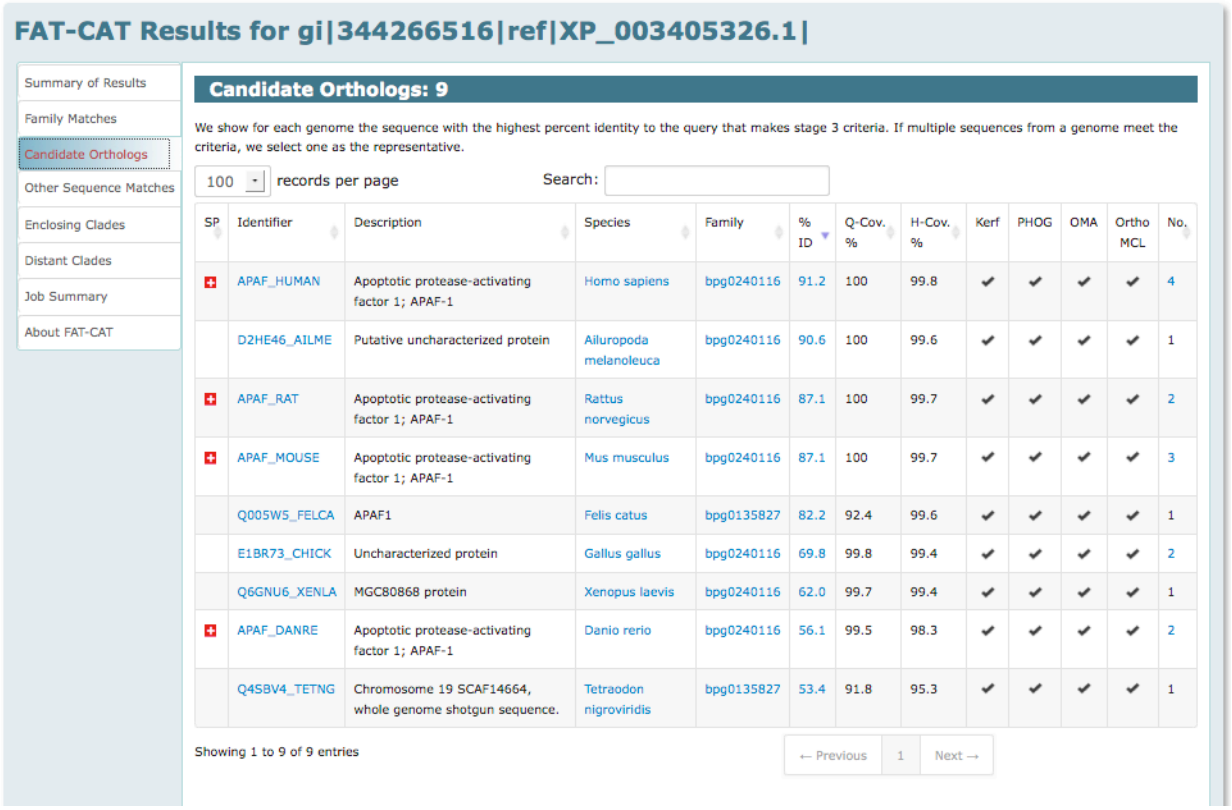


Figure 22. FAT-CAT results page, displaying candidate orthologs. In this example, nine candidate orthologs are identified; all are supported by all four orthology methods used. If multiple sequences are found from the same genome with the same sequence identity to the query, we pick one as the representative; if one of the cluster is in the SwissProt database, we use that as the representative, else we pick the one with the highest sequence identity.

The Candidate Orthologs data table includes data to help you evaluate the support for each proposed ortholog, and to explore data associated with that ortholog:

- Alignment statistics are provided, including the percent identity, query coverage and hit coverage. Clicking on the value in the %ID column will display the alignment between the query and candidate ortholog.
- Tool-tipping the species will display the common name for that species; clicking will bring you to the NCBI taxonomy.
- Clicking on the sequence identifier will bring you to the UniProt page for that sequence. SwissProt sequences are indicated by the red flag at far left.
- The bpg accession is linked to the family containing the enclosing clade in which the candidate ortholog was found.

4.5 Other Sequence Matches

SP	Identifier	Description	Species	Family	% ID	Q-Cov. %	H-Cov. %	Kerf	PHOG	OMA	Ortho MCL	No.
	Q80VR5_MOUSE	Apaf1 protein; Apoptotic protease-activating factor 1	Mus musculus	bpg0175819	82.6	20.7	100	✔	✔	✔	✔	1
	D3ZA56_RAT	Uncharacterized protein	Rattus norvegicus	bpg0240116	81.2	99.7	99.4	✔	✔	✔	✔	1
	Q8HXQ8_HORSE	Apoptotic protease activating factor 1	Equus caballus	bpg0175819	75.0	6.7	100	✔	✔	✔	✔	1
	A8WH04_XENTR	LOC100127738 protein	Xenopus tropicalis	bpg0175819	66.7	26.5	100	✔	✔	✔		1
	Q1JPV6_DANRE	Apoptotic protease activating factor 1	Danio rerio	bpg0135827	60.0	37.1	99.6	✔	✔	✔	✔	1
	Q10BZ9_ORYSJ	Transducin family protein, putative, expressed; cDNA clone:J023024A21, full insert sequence	Oryza sativa subsp. japonica	bpg0218243	45.3	20.6	95.9	✔				1
	D5G214_PODAS	HET-R	Podospora anserina	bpg0237571	44.2	20	98.8	✔				1
	D5G224_PODAS	NWD1	Podospora anserina	bpg0224070	42.9	23.6	100	✔	✔			1
	Q8Z054_NOSS1	WD-40 repeat protein	Nostoc sp. (strain PCC 7120 / UTEX 2576)	bpg0211459	42.6	23.4	95.7	✔	✔			1

Figure 23. FAT-CAT results page, displaying other sequence matches. Sequences displayed on this page have been rejected as candidate orthologs due to failing one or more orthology criteria.

In this example, 199 sequences were found in one or more enclosing clades but rejected as candidate orthologs.

Several sequences are labeled as APAF1 sequences, which might lead one to assume they are orthologs. However, the alignments are either partial (over only a subregion) or the sequence identity falls below the minimum allowed. In other cases, another sequence from the same genome has significantly higher sequence identity so was selected as the "true" ortholog, and the second-ranked sequence from that genome was rejected. Some of these rejected sequences are clearly isoforms or different gene models for the same underlying gene, as shown on the next page.

Why are some sequences rejected as orthologs despite having high sequence identity?

Sequences in an enclosing clade that are listed in Other Sequence Matches are typically rejected as orthologs due to not meeting alignment criteria. In this example, four of the top five fail the query coverage requirements, i.e., they align along only a subregion. The second sequence listed, from *Rattus norvegicus*, has 81% sequence identity and near-perfect overlap (for both the query and hit). However, another sequence from the same genome with higher sequence identity (87%) and 100% overlap (bi-directional, to both the query and hit) was selected as an ortholog and listed in the Candidate Orthologs tab. See next figure.

Note the rat ortholog selected and compare with the rat sequence rejected due to lower sequence identity and placed in Other Sequence Matches. (Both are presumably from the same gene, but APAF_RAT, from SwissProt, represents the best gene model for this gene, so is selected as the ortholog.)

Other Sequence Matches: 199

Sequences on this page are drawn from the Enclosing Clades of candidate orthologs but fail one or more criteria for orthology.

100 records per page Search:

SP	Identifier	Description	Species	Family	% ID	Q-Cov. %	H-Cov. %	Kerf	PHOG	OMA	Ortho MCL	No.
	Q80VR5_MOUSE	Apaf1 protein; Apoptotic protease-activating factor 1	Mus musculus	bpg0175819	82.6	20.7	100	✔	✔	✔	✔	1
	D3ZA56_RAT	Uncharacterized protein	Rattus norvegicus	bpg0240116	81.2	99.7	99.4	✔	✔	✔	✔	1

Close-up view of top-ranked sequences in the "Other Sequence Matches" tab.

Candidate Orthologs: 9

We show for each genome the sequence with the highest percent identity to the query that makes stage 3 criteria. If multiple sequences from a genome meet the criteria, we select one as the representative.

100 records per page Search:

SP	Identifier	Description	Species	Family	% ID	Q-Cov. %	H-Cov. %	Kerf	PHOG	OMA	Ortho MCL	No.
✚	APAF_HUMAN	Apoptotic protease-activating factor 1; APAF-1	Homo sapiens	bpg0240116	91.2	100	99.8	✔	✔	✔	✔	4
	D2HE46_AILME	Putative uncharacterized protein	Ailuropoda melanoleuca	bpg0240116	90.6	100	99.6	✔	✔	✔	✔	1
✚	APAF_RAT	Apoptotic protease-activating factor 1; APAF-1	Rattus norvegicus	bpg0240116	87.1	100	99.7	✔	✔	✔	✔	2

Close up view of the top-ranked orthologs, including APAF_RAT.

4.6 How does FAT-CAT assign functional annotations to a query?

The **Functional Annotations** section of the Summary of Results displays predicted functions for the query derived from putative orthologs. We derive a weighted consensus over the UniProt description of the protein's function, weighting close orthologs and manually curated annotations more than distant orthologs and annotations derived automatically. We also display the union over all Gene Ontology annotations (biological process, molecular function and cellular location) along with the best Evidence Code available for that annotation; clicking on the annotation will bring up a list of orthologs having that annotation.

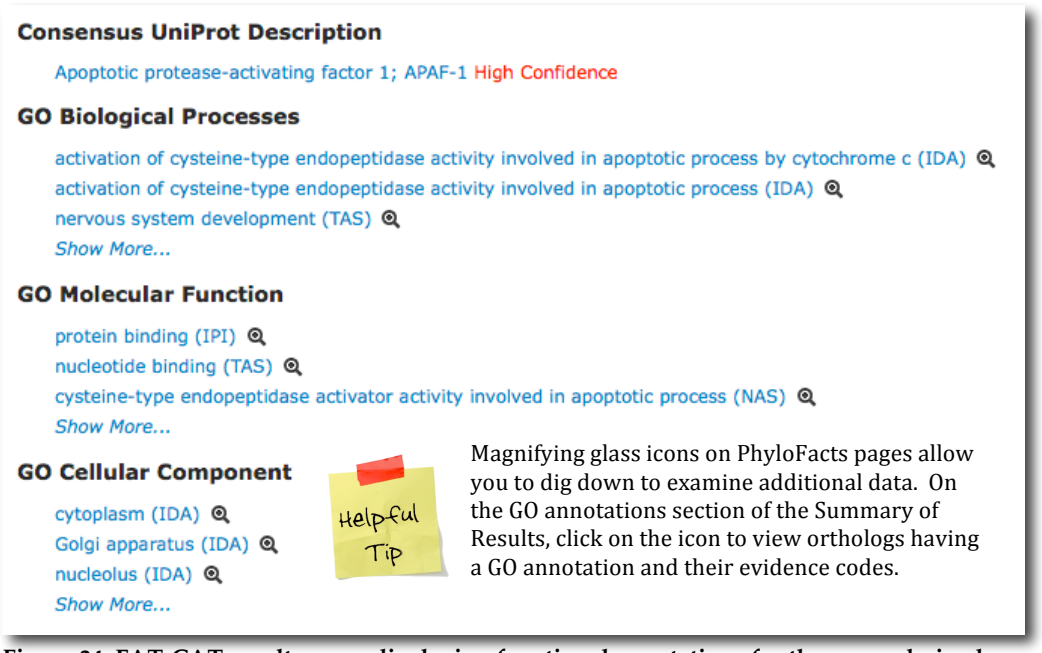
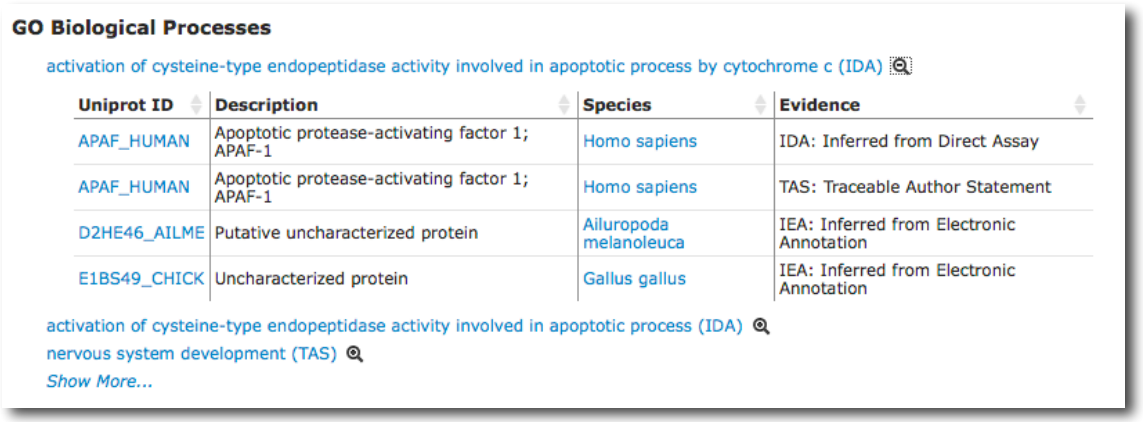


Figure 24. FAT-CAT results page, displaying functional annotations for the query derived from orthologs..



Clicking on the magnifying glass next to a GO annotation shows the orthologs having that annotation along with the evidence codes.

References Cited

1. Eisen, J.A., *Phylogenomics: improving functional predictions for uncharacterized genes by evolutionary analysis.* Genome Res, 1998. **8**(3): p. 163-7.
2. Bork, P. and E.V. Koonin, *Predicting functions from protein sequences--where are the bottlenecks?* Nat Genet, 1998. **18**(4): p. 313-8.
3. Galperin, M.Y. and E.V. Koonin, *Sources of systematic error in functional annotation of genomes: domain rearrangement, non-orthologous gene displacement and operon disruption.* In Silico Biol, 1998. **1**(1): p. 55-67.
4. Gerlt, J.A. and P.C. Babbitt, *Can sequence determine function?* Genome Biol, 2000. **1**(5): p. REVIEWS0005.
5. Gilks, W.R., et al., *Modeling the percolation of annotation errors in a database of protein sequences.* Bioinformatics, 2002. **18**(12): p. 1641-9.
6. Brenner, S.E., *Errors in genome annotation.* Trends Genet, 1999. **15**(4): p. 132-3.
7. Jones, C.E., A.L. Brown, and U. Baumann, *Estimating the annotation error rate of curated GO database sequence annotations.* BMC Bioinformatics, 2007. **8**: p. 170.
8. Green, M.L. and P.D. Karp, *Genome annotation errors in pathway databases due to semantic ambiguity in partial EC numbers.* Nucleic Acids Res, 2005. **33**(13): p. 4035-9.
9. Mi, H., et al., *Assessment of genome-wide protein function classification for Drosophila melanogaster.* Genome Res, 2003. **13**(9): p. 2118-28.
10. Schnoes, A.M., et al., *Annotation error in public databases: misannotation of molecular function in enzyme superfamilies.* PLoS Comput Biol, 2009. **5**(12): p. e1000605.
11. Gilks, W.R., et al., *Percolation of annotation errors through hierarchically structured protein sequence databases.* Math Biosci, 2005. **193**(2): p. 223-34.
12. Sjolander, K., *Phylogenomic inference of protein molecular function: advances and challenges.* Bioinformatics, 2004. **20**(2): p. 170-9.
13. Delsuc, F., H. Brinkmann, and H. Philippe, *Phylogenomics and the reconstruction of the tree of life.* Nat Rev Genet, 2005. **6**(5): p. 361-75.
14. Venter, J.C., et al., *The sequence of the human genome.* Science, 2001. **291**(5507): p. 1304-51.
15. Brown, D.P., N. Krishnamurthy, and K. Sjolander, *Automated protein subfamily identification and classification.* PLoS Comput Biol, 2007. **3**(8): p. e160, PMCID: PMC1950344.
16. Mi, H., et al., *PANTHER version 6: protein sequence and function evolution data with expanded representation of biological pathways.* Nucleic Acids Res, 2007. **35**(Database issue): p. D247-52.
17. Glanville, J.G., et al., *Berkeley Phylogenomics Group web servers: resources for structural phylogenomic analysis.* Nucleic Acids Res, 2007. **35**(Web Server issue): p. W27-32.
18. Krishnamurthy, N., et al., *PhyloFacts: an online structural phylogenomic encyclopedia for protein functional and structural classification.* Genome Biol, 2006. **7**(9): p. R83.
19. Sjolander, K., *Getting Started in Structural Phylogenomics.* PLoS Comput Biol, 2010. **6**(1).
20. Fitch, W.M., *Distinguishing homologous from analogous proteins.* Syst Zool, 1970. **19**(2): p. 99-113.

21. Dessimoz, C., et al., *Detecting non-orthology in the COGs database and other approaches grouping orthologs using genome-specific best hits.* Nucleic Acids Res, 2006. **34**(11): p. 3309-16.
22. Zmasek, C.M. and S.R. Eddy, *RIO: analyzing proteomes by automated phylogenomics using resampled inference of orthologs.* BMC Bioinformatics, 2002. **3**: p. 14.
23. Sonnhammer, E.L. and E.V. Koonin, *Orthology, paralogy and proposed classification for paralog subtypes.* Trends Genet, 2002. **18**(12): p. 619-20.
24. Datta, R.S., et al., *Berkeley PHOG: PhyloFacts orthology group prediction web server.* Nucleic Acids Res, 2009. **37**(Web Server issue): p. W84-9.
25. Krishnamurthy, N., D. Brown, and K. Sjolander, *FlowerPower: clustering proteins into domain architecture classes for phylogenomic inference of protein function.* BMC Evol Biol, 2007. **7 Suppl 1**: p. S12, PMCID: PMC1796606.
26. Katoh, K. and H. Toh, *Recent developments in the MAFFT multiple sequence alignment program.* Brief Bioinform, 2008. **9**(4): p. 286-98, PMID: 18372315.
27. Datta, R.S., et al., *Berkeley PHOG: PhyloFacts orthology group prediction web server.* Nucleic Acids Res, 2009. **37(Web Server issue):** : p. W84-W9, PMCID: PMC2703887.
28. Bateman, A., et al., *The Pfam protein families database.* Nucleic Acids Res, 2004. **32**(Database issue): p. D138-41.
29. Sjolander, K., et al., *Ortholog identification in the presence of domain architecture rearrangement.* Brief Bioinform, 2011.
30. Li, H., et al., *TreeFam: a curated database of phylogenetic trees of animal gene families.* Nucleic Acids Res, 2006. **34**(Database issue): p. D572-80.