Berkeley PHOG: PhyloFacts Orthology Group Prediction Web

Server Supplement 1: Description of the PHOG Algorithm

Ruchira Datta[1] and Kimmen Sjölander[*,1,2,3]

[1]QB3 Institute, University of California, Berkeley, [2] Department of Bioengineering, University of California, Berkeley, and [3] Department of Plant and Microbial Biology, University of California, Berkeley

We describe here the algorithm used to predict orthologs. Consider a phylogenetic gene tree with leaves corresponding to sequences in different taxa (species). The same taxon (species) can be found at many different leaves in the gene tree (i.e., the mapping from the gene tree to the species tree is many-to-one). Let Q be one of these proteins, belonging to taxon T. We would like to know for each other protein P in the tree whether it is an ortholog of Q,, i.e., whether the most recent common ancestor of P and Q represents a speciation event.

The most rigorous method for predicting orthologs is *tree reconciliation*, in which the species tree is overlaid on the gene tree in order to label each node as a speciation event or a duplication event. We discuss how to derive several kinds of "ortholog", in other words, to apply a few different methods for predicting orthologs using this definition.

Definition 1: A protein P in species S is a *nearest-neighbor ortholog* of Q in S, if the distance between P and Q in the tree is shortest among all proteins from S that are present in the tree.

Definition 2: The proteins P in species S and Q in species T are *RNN (reciprocal-nearest neighbor) orthologs*, if P is the nearest-neighbor ortholog of Q in S and Q is the nearest-neighbor ortholog of P in T.

Definition 3. A *PHOG-S (phylogenomic orthologous group)* is a maximal subtree such that every protein in that subtree is an RNN ortholog of every other protein.

While tree reconciliation is a computationally expensive process whose accuracy strongly depends on that of the species tree, PHOGs can be inferred efficiently and do not depend on a species tree. We briefly explain the issue of inparalogs before describing the algorithm to infer PHOGs.

Definition 4. Two proteins in a protein family tree are defined as *inparalogs* if they are from the same species and are separated only by duplication events, that is, they are the result of duplication that occurred after the speciation resulting in their current species. These are also termed *ultraparalogs* by Zmasek & Eddy (BMC Bioinformatics, 2002).

Another common situation complicating tree reconciliation and causing an overestimate of the number of duplication events is the presence of redundant entries in a sequence database for the same gene, i.e., with sequence differences arising from sequencing errors, population variation, and so forth. In practice, such sequences cannot be distinguished from true inparalogs without genomic context information. For brevity, we refer to all such sequences as inparalogs, although they may not be inparalogs in fact.

Definition 5. If two sequences P1 and P2 from species S are inparalogs, and each one is an ortholog of the sequence Q from species T, then P1 and P2 are *co-orthologs* of Q.

If a subtree contains only sequences from one species, i.e., it is *pure* with respect to that species, we can infer that all the sequences in that subtree are inparalogs.

Definition 6. An inparalogous subtree is a subtree containing only sequences from one species. A mixed subtree is a subtree containing sequences from more than one species.

We will allow inparalogous subtrees to be part of our PHOGs, so we modify the definition of a PHOG as follows.

Definition 7. A PHOG is a maximal subtree such that for every pair of proteins P with species S and Q with species T in the PHOG where S and T are distinct, P has an inparalog P* (possibly P itself) and Q has an inparalog Q* (possibly Q itself) such that P* and Q* are RNN orthologs. In short, a PHOG is a maximal subtree, all of whose leaves are RNN orthologs or co-orthologs of each other.

We start by constructing an order in which to visit all the nodes of the tree, the *breadth-first visit order*. In breadth-first search of a tree, one puts the nodes to be visited onto a queue. One starts by pushing the root onto the front of the queue. Every time one pops a node off the front of the queue, one pushes all its children onto the back of the queue. Each node is visited exactly once, so breadth-first search takes O(n) time, where n is the number of nodes in the tree.

Note that in our application, we will need to use this order multiple times, backwards and forwards. So we use a vector instead of a queue. Instead of popping nodes off the front, we simply increment our index into the vector past them. This visit order has the important property that each parent is visited before any of its children.

To infer the inparalogous subtrees, we visit each node in the *reverse* breadth-first visit order. This ensures that before we visit a node, we have visited all its children. We mark every leaf with the species of the protein at that leaf. When we visit a node, if all its children have been marked with the same species, we mark it with that species. Otherwise, we mark each of its children that is marked with some species as maximal. At the end, each of these maximal nodes defines an inparalogous subtree.

Now we can infer the nearest-neighbor orthologs, RNN orthologs, and PHOGs in time O(nm), where n is the number of nodes in the tree and m is the number of species (counting only those species with at least two leaves in the tree), using a dynamic programming algorithm. We will mark each node with the closest leaf in each taxon, along with the distance from the node to that leaf. Recall that the distance between a pair of nodes is the sum over the edge lengths along the shortest path joining the two nodes, not the number of edges on that path. We start by recording that each leaf is at distance 0 from the taxon of the protein at that leaf. Then we start visiting the nodes in reverse breadth-first visit order. For each node N, for each species S, suppose P has been recorded as the closest protein in S to N, at distance d_P. Let d_l be the length of the branch between N and its parent M and suppose R has been recorded as the closest protein in S to M, at distance d_R. If $d_P + d_l < d_R$, then we update M, recording that its closest protein in S is P, at a distance $d_P + d_l$. To ensure a consistent choice among inparalogs at tree distance 0 from each other, if $d_P + d_l = d_R$, we will also update M, recording that its closest protein in S is P, if P < R in some arbitrary ordering that was fixed in advance.

After we have done this, we visit each node again, in the breadth-first visit order. This time, when we visit a node, instead of updating its parent we update each of its children, checking to see whether the closest protein to the parent is closer to the child than the one (if any) currently recorded as closest to the child. Since every shortest path between leaves of a tree is V-shaped, first going upwards from nodes to their parents and then going downwards from nodes to their children, when we are done we will have checked all possibilities and so will have found the closest protein in each taxon to every node of the tree. In the case of a leaf, these are the nearest-neighbor orthologs of the protein at that leaf in each other taxon.

Now we again traverse the tree in reverse breadth-first visit order. For each node, for each taxon, we record whether one of the following two conditions is true: the node is inparalogous with that taxon, or the same protein from that taxon is closest to that node and to all of its children. If so, then we mark that node as *sequence pure* with respect to that taxon. (In particular, each leaf is sequence pure with respect to every taxon.) If not, we record any of its children that is sequence pure as maximal for that sequence. If any of these children is mixed, then we record that the node has a mixed maximal descendant. We do the same if any of the children has a mixed maximal descendant. If a node is sequence pure with respect to a taxon S, then the protein P from taxon S which is closest to that node is also closest to all of its descendants.

When we are done, every maximal node M which has no mixed maximal descendants defines a PHOG.

Proof: First, suppose there are no inparalogous subtrees under M. Then M is sequence pure with respect to each of the taxa of its descendants. If it were not sequence pure with respect to the taxon S of the protein P at some leaf L, let N be the maximal node between M and L which is sequence pure with respect to S. Then N is a mixed maximal descendant of M, contrary to our assumption. The case including inparalogous subtrees reduces to the previous one by collapsing each inparalogous subtree to a single node and assigning it the protein P* from that species which is closest to it in the original tree. On the other hand, the parent K of M must not be sequence pure with respect to all the taxa that M is, otherwise M would not be maximal. So there must be some taxon T such that Q1 is the closest protein to M in T, but Q2 is the closest protein to K in T. The shortest path from Q1 to M cannot go through K, so Q1 must be a descendant of M. But it is not an RNN ortholog of any descendant of K which is not a descendant of M, because Q2 is closer to these. This shows that M is a maximal subtree such that all of its leaves are RNN orthologs or co-orthologs of each other.

In the algorithm so far described, PHOG-S, we only mark a maximal node as a PHOG if it has no mixed maximal descendants. However, see Figure 1, below, where technical challenges in tree estimation arise among very closely related species.

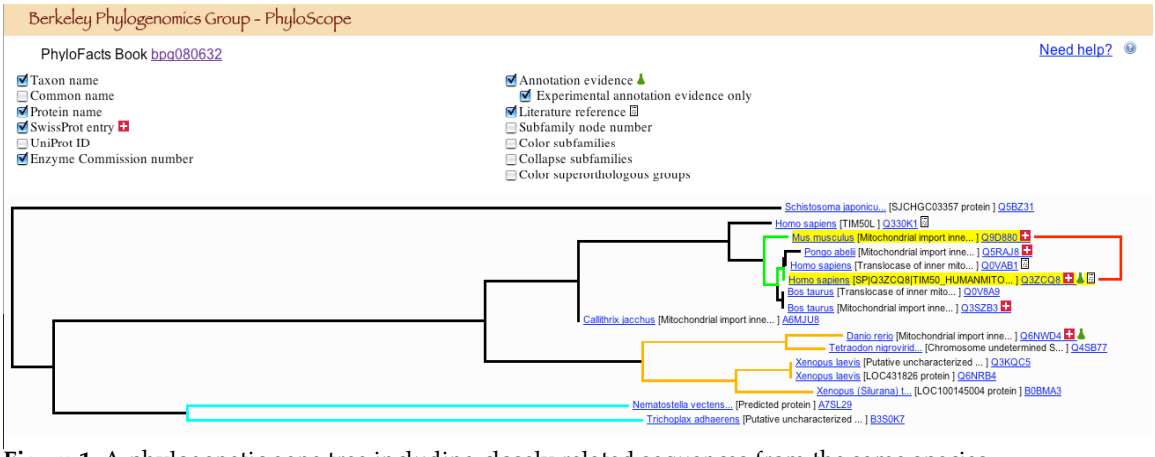

Figure 1. A phylogenetic gene tree including closely related sequences from the same species.

In this tree the human sequence Q3CZQ8 is in a subtree T containing a smaller subtree T' with another human sequence, Q0VAB1, and an orangutan sequence, Q5RAJ8. Because T' contains sequences from multiple species, it is mixed (i.e., not inparalogous). Thus it is a 2-sequence PHOG. So the PHOG containing Q3CZQ8 is the leaf itself, a 1-sequence PHOG. Thus PHOG-S makes no orthology predictions for this sequence.

The sequences in T are very close to each other. If we would like to find orthologs in a more distant species, such as mouse, we may not be concerned about the fine branching order between very closely related sequences, such as the orangutan and human sequences. For this reason, we introduce a variant algorithm PHOG-T to which we can pass a parameter, the threshold T. For each maximal node M and for each species S for which it is maximal, let d_S be the distance from M to the nearest sequence in S and let n_S be the distance from M to the next-nearest sequence in S. Let $t'_{M,S} = (d_S + n_S)/2$, and let $t'_M = \max_S t'_{M,S}$ be the *duplication distance*. Intuitively, since M is maximal, we consider that a duplication may have occurred above M, and we approximate its distance from the leaves by the midpoint of the span between the two sequences. Next, we let t_M be the maximum of t'_M and t'_N for every maximal node N which is a descendant of M. Finally, we consider as PHOGs only those maximal nodes M for which $t_M >= T$, the threshold parameter. If such a node M contains descendant nodes N also satisfying this condition, then we define the PHOG M to be the complement of all such subtrees N within the subtree M. The members of an orthology group predicted by PHOG-T are no longer predicted to be super-orthologous, but simply orthologous. However, since the PHOG-Ts as we have defined them form disjoint subsets of the tree, the PHOG-T relationship is still transitive.

Intuitively, by thresholding above a certain "duplication distance" we are choosing to consider descendants within the same species of a duplication event more recent than that to be inparalogs/co-orthologs, regardless of the detailed tree structure under the node corresponding to the duplication event. If we are actually interested in orthology with a query sequence in a relatively distant species, then the most recent common ancestor of the query sequence and these "co-orthologs" is likely to be less recent than this duplication event (i.e., it is likely to be an ancestral node of the duplication event as well), so indeed these sequences would be co-orthologs to the query sequence by the usual definition.

In the PHOG-O variant, we label as duplication events each node which is a parent of one of the maximal nodes. We predict two sequences in the tree whose most recent common ancestor is not such a duplication event to be orthologous.