Syst. Biol. 55(5):769–773, 2006
Copyright © Society of Systematic Biologists
ISSN: 1063-5157 print / 1076-836X online
DOI: 10.1080/10635150600981604

Phylogenetic Diversity within Seconds

BUI QUANG MINH, STEFFEN KLAERE, AND ARNDT VON HAESELER

Center for Integrative Bioinformatics, Vienna, Max F. Perutz Laboratories, University of Vienna, Medical University of Vienna, Veterinary University of Vienna, Dr.-Bohr-Gasse 9/6, A-1030 Vienna, Austria; E-mail: arndt.von.haeseler@univie.ac.at (A.v.H.)

Abstract.—We consider a (phylogenetic) tree with n labeled leaves, the taxa, and a length for each branch in the tree. For any subset of k taxa, the phylogenetic diversity is defined as the sum of the branch-lengths of the minimal subtree connecting the taxa in the subset. We introduce two time-efficient algorithms (greedy and pruning) to compute a subset of size k with maximal phylogenetic diversity in $O(n \log k)$ and $O[n + (n-k)\log(n-k)]$ time, respectively. The greedy algorithm is an efficient implementation of the so-called greedy strategy (Steel, 2005; Pardi and Goldman, 2005), whereas the pruning algorithm provides an alternative description of the same problem. Both algorithms compute within seconds a subtree with maximal phylogenetic diversity for trees with 100,000 taxa or more. [Biodiversity conservation; Comparative genomics; Greedy algorithm; Phylogenetic diversity; Phylogenetic tree; Pruning algorithm.]

Recently, Steel (2005) and Pardi and Goldman (2005) have shown that being greedy works if one is interested in selecting k taxa from a phylogenetic tree that maximize the phylogenetic diversity. The term *phylogenetic diversity* (*PD*) was coined by Faith (1992) to provide an effective measure of the diversity of a group of taxa. The optimal *PD* describes the amount of diversity embraced by a properly chosen subset of taxa. Faith (1992) applied *PD* to place conservation priorities on different taxa, where the taxa to protect reflect a certain value of taxonomic diversity. Thus, some measurable indicator of biodiversity defined on different scales (taxa, group of taxa, ecosystems, etc.) is assigned to the corresponding systematic categories. With the advent of molecular genetics, evolutionary divergence on the genomic level may also serve this purpose (Pardi and Goldman, 2005).

For the following, the precise nature of the measure of phylogenetic diversity is not relevant (cf. Humphries et al., 1995; Williams and Araujo, 2002, for a discussion on diversity measures). Phylogenetic diversity should simply describe the overall value of a group of taxa either in terms of genetic diversity, regional diversity, or social diversity. Moreover, it is required that these measures can be mapped onto a phylogenetic tree in a way that the branches of the tree receive non-negative weights.

The problem is then as follows: From a tree with n taxa, one wants to identify k taxa that retain the maximal phylogenetic diversity, therefore taking into account the fact that due to restricted resources only a certain percentage of the taxa can be sustained. Steel (2005) and Pardi and Goldman (2005) have proven that a greedy approach yields the optimal set with respect to *PD*. The greedy strategy repeatedly selects the taxon that adds the most divergence to the already chosen set of taxa. The procedure is repeated until k taxa are found. Both proofs apply—directly or indirectly—the theory of weighted matroids and greedy algorithms (Korte et al., 1991). From this theory it follows that an algorithm with time complexity $O(n \log n)$ is possible.

In the following, we will suggest a time-efficient *greedy phylogenetic diversity algorithm* (*gPDA*). Moreover, a different but easier to implement algorithm, the *pruning phylogenetic diversity algorithm* (*pPDA*) will be introduced. Both algorithms compute the optimal k set for large phylogenies within seconds.

NOTATION

Following Steel (2005), we call $\mathcal{T}$ an unrooted phylogenetic X-tree; that is, a tree with leaf set X of taxa and whose remaining interior nodes are of degree at least three. $\mathcal{V}$ denotes the set of all nodes of $\mathcal{T}$ and $\mathcal{E}$ the collection of edges or branches. λ denotes the edge-weight function that assigns to each edge $e = (v, w)$, $(v, w \in \mathcal{V})$ of $\mathcal{T}$ a (non-negative) branch length $\lambda(v, w) \geq 0$.

A path $\mathcal{P}(a, b)$ denotes the collection of distinct nodes $a = v_0, v_1, \ldots, v_{m+1} = b$ in a tree such that v_i, v_{i+1} are adjacent; i.e., connected by an edge. The sum of the edge weights of all edges along the path between two nodes a and b denotes their distance $d(a, b)$ in the tree.

To describe the algorithms, it will be handy to root $\mathcal{T}$ at a node r. Then the remaining leaves are descendents from r. Thus, for each node $v \in \mathcal{V}$ the set $L_{\max}(v)$ is well defined and denotes the descendant(s) farthest away from v. For the sake of clarity, we abbreviate the distance $d[v, L_{\max}(v)]$ as $d_{\max}(v)$.

For a subset W of X we consider $\mathcal{T}|W$, the induced phylogenetic W-tree, that connects all taxa in W according to $\mathcal{T}$. Finally, λ_W assigns to each edge e of $\mathcal{T}|W$ the sum of the $\lambda(e)$ values over those edges in $\mathcal{T}$ along the path that corresponds to the new edge e. The *phylogenetic diversity* of W, denoted *PD(W)*, is then

$$PD(W) = \sum_{e} \lambda_W(e),$$

where the summation is over all edges e in the tree $\mathcal{T}|W$ (see Steel, 2005).

THE TIME-EFFICIENT GREEDY ALGORITHM: gPDA

We briefly describe the implementation of *gPDA*. The phylogenetic tree $\mathcal{T}$ together with its weight-function and the size of k define the input of the algorithm. We want to determine the collection W of k taxa with maximal phylogenetic diversity. In the following, we describe the algorithm for trees with interior nodes of degree three. However, the implementation works for trees with finite interior node degree of at least three. *gPDA* splits in two steps.

The *initial step* starts with the computation of the longest path in $\mathcal{T}$. This can be achieved in $O(n)$ time by applying a depth-first search (DFS) (cf. Cormen et al., 2001). The algorithm starts at an arbitrary leaf c and determines the leaf a furthest away from c in $\mathcal{T}$. It is easy to show that a is one of the endpoints of the longest path in $\mathcal{T}$. We root the tree at a and based on this root compute for all interior nodes v_i the distance $d_{\max}(v_i)$ and the associated set $L_{\max}(v_i)$. This is again a DFS procedure; i.e., has complexity $O(n)$. Figure 1A displays the result of this procedure for a tree with five taxa. The longest path in the tree has distance 20. Thus, the set W is equal to $\{a, b\}$.

To extend W, we note that for each leaf c in $\mathcal{V} - W$, exactly one node v_i ($i = 1, \ldots, m$) in $\mathcal{P}(a, b)$ acts as ancestor; i.e., v_i is the node where the paths $\mathcal{P}(a, c)$ and $\mathcal{P}(a, b)$ split. Obviously, one selects the leaf that is farthest away from its ancestor in $\mathcal{P}(a, b)$. To this end, we generate an ordered list $\mathcal{S}$ with respect to $d_{\max}$ that contains at most $k - 2$ nodes $v_1, v_2, \ldots, v_{k-2}$ from the path set $\mathcal{P}(a, b)$. In $\mathcal{S}$ the nodes are ordered in descending order according to $d_{\max}$; i.e., the following holds:

$$d_{\max}(v_{i_1}) \geq d_{\max}(v_{i_2}) \geq \cdots \geq d_{\max}(v_{i_{k-2}}).$$

Before generating $\mathcal{S}$, we must update for each v_i on $\mathcal{P}(a, b)$ the set $L_{\max}(v_i)$ and $d_{\max}(v_i)$ by choosing a leaf c with maximal distance to v_i such that $\mathcal{P}(v_i, c)$ does not have an edge in common with the path $\mathcal{P}(a, b)$. For each node v_i this update can be done in constant time. If $\mathcal{P}(a, b)$ contains more than $k - 2$ nodes and $\mathcal{S}$ has already $k - 2$ elements, then a new node v from $\mathcal{P}(a, b)$ is only added to $\mathcal{S}$ if $d_{\max}(v) > d_{\max}(v_{i_{k-2}})$. The node $v_{i_{k-2}}$ is subsequently deleted from $\mathcal{S}$ and v is inserted at its appropriate position in $\mathcal{S}$. This step takes $O(n \log k)$ time in the worst case.

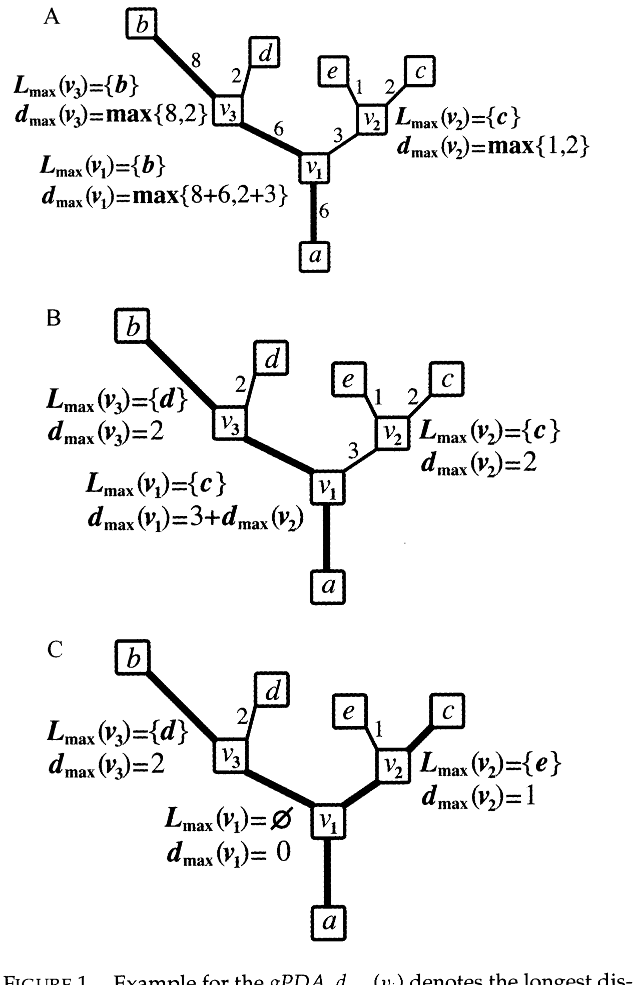

FIGURE 1. Example for the *gPDA*. $d_{\max}(v_i)$ denotes the longest distance between v_i and its descending taxa, and $L_{\max}(v_i)$ denotes the set of taxa with distance $d_{\max}(v_i)$ to v_i. (A) Result of the greedy strategy after selecting the longest path (bold lines). (B) Updating nodes on the longest path in the initial step. (C) Adding leaf c to W and updating the nodes on the partial tree.

Figure 1B displays the result of this update for the five-taxon tree. Here, we obtain $\mathcal{S} = (v_1, v_3)$, because $d_{\max}(v_1) = 5 > 2 = d_{\max}(v_3)$. This update procedure will be invoked repeatedly in the following step of *gPDA*.

Having defined W and a sorted list $\mathcal{S}$ we can enter the core of the algorithm, the *greedy step*.

We add a leaf c from $L_{\max}(v_{i_1})$ to W and delete v_{i_1} from $\mathcal{S}$. Then we update the maximal distances and leaves for all nodes on the path $\mathcal{P}(v_{i_1}, c)$ as described for the path $\mathcal{P}(a, b)$. No updates are necessary for interior nodes already in $\mathcal{S}$. Figure 1C illustrates this second update for the example tree with $W = \{a, b, c\}$ and $\mathcal{S} = \{v_3\}$. v_1 and v_2 are updated, whereas v_3 remains unchanged.

Subsequently, the elements w of the path $\mathcal{P}(v_{i_1}, c)$ are inserted into the ordered list $\mathcal{S}$ according to their distance $d_{\max}(w)$ if $d_{\max}(w) \geq d_{\max}(v_{i_{k-2}})$. In the sample tree v_2 is added and thus $\mathcal{S} = \{v_3, v_2\}$. This completes the greedy step. The greedy step is repeated until W contains k taxa.

To determine the complexity of *gPDA*, recall that computing the longest path and identifying taxa a and b in the initial step consumes $O(n)$ time. The time requirement to generate and update $\mathcal{S}$ is more subtle to establish. Because W will eventually contain k taxa, the cardinality of $\mathcal{S}$ is never larger than $k - 2$. At any time, the $k - 2$ nodes in $\mathcal{S}$ are the most promising for $\mathcal{T}|W$. An insertion of an interior node into $\mathcal{S}$ requires $O(\log k)$ time, because $\mathcal{S}$ is implemented as a red-black search tree data structure (e.g., Cormen et al., 2001, chap. 13). Each interior node is inserted in $\mathcal{S}$ at most once during the $k - 2$ greedy steps. Because a bifurcating tree with n taxa has $n - 2$ interior nodes, generating and updating $\mathcal{S}$ takes $O(n \log k)$ time. Therefore, the overall worst-case time complexity of *gPDA* is $O(n \log k)$.

AN EFFICIENT PRUNING ALGORITHM: pPDA

Easier to implement is the *pruning phylogenetic diversity algorithm* (*pPDA*), a special application of the so-called worst-out greedy algorithm (Korte et al., 1991). Here, we start with the full tree of n taxa. Based on the length $\lambda(v, x)$ of an exterior edge leading to a leaf $x \in X$, we compute a sorted list $\mathcal{S}$ of the taxa, arranged in ascending order. This completes the *initial step* of the algorithm.

In the following $n - k$ iterations (*pruning steps*), the first taxon s_1 in the list is deleted from $\mathcal{S}$. The degree of the node v that forms the branch (v, s_1) is decreased by one. If the new degree of v equals two, then the incident edges of v are joined and the branch length of the new edge is the sum of the lengths of the joined edges. Moreover, if the new edge is connected to a leaf, the branch length of the leaf is updated. Subsequently, the leaf is put at its appropriate position in $\mathcal{S}$.

After $n - k$ pruning steps, $\mathcal{S}$ contains k taxa that constitute the set W with maximal phylogenetic diversity. It is straightforward to prove that *pPDA* provides trees with maximal phylogenetic diversity. Its optimality follows immediately from the "strong exchange" property of *PD* (Steel, 2005). This algorithm is so simple that it can be carried out on a piece of paper. We conclude the section with the discussion of its complexity.

At each pruning step at most one taxon must be repositioned in $\mathcal{S}$. We also note that the new position of the taxon is always further down the sorted list, because the length of an incident branch always increases. Thus to complete $n - k$ pruning steps, $\mathcal{S}$ needs to store in the worst case $2(n - k)$ taxa. Therefore, in the initial step, the selection of those taxa can be done in $O(n)$ time (e.g., Cormen et al., 2001). Then we only have to sort the selected taxa in $O[(n - k)\log(n - k)]$ time, because $\mathcal{S}$ is implemented as a red-black search tree (e.g., Cormen et al., 2001). Finally, repositioning a taxon in the pruning step needs at most $O[\log(n - k)]$ steps. Thus, the complexity of the $n - k$ pruning steps amounts to $O[(n - k)\log(n - k)]$. This results in an overall complexity of *pPDA* of $O[n + (n - k)\log(n - k)]$.

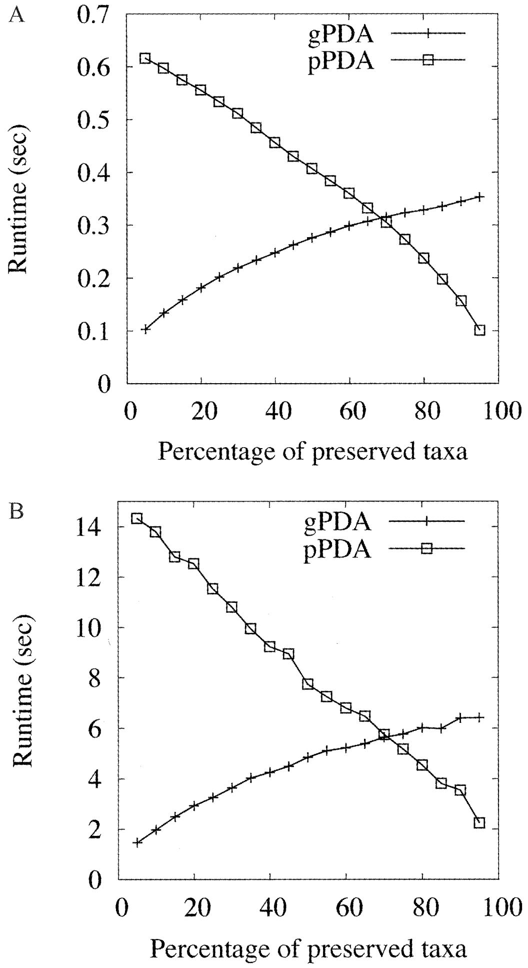

FIGURE 2. Comparison of computing times of *gPDA* and *pPDA*. Each point represents the average run time from 100 runs for n =100,000 (A) and n = 1,000,000 taxa (B), respectively. Subset sizes ranging from $k = 5\% \cdot n, \ldots, 95\% \cdot n$.

RUN TIME ANALYSIS

We conducted computer simulations to test the wall-clock computing time of *gPDA* and *pPDA*. Simulations were performed on a 2-GHz AMD Opteron 246 with 2-GByte RAM. Both algorithms were so fast that only for huge trees with more than 100,000 taxa was a substantial difference in the performance observed. Therefore, we will only compare the results for $n = 100{,}000$ and 1,000,000 taxa, respectively. The computing times (in seconds) in Figure 2 are based on average times from 100 random trees generated under the Yule-Harding model (Harding, 1971) for each combination of the pair (n, k). The branch lengths are randomly drawn from the interval (0, 1). The size k of W was varied from 5% to 95% of the n taxa in the tree.

For the $n = 10^5$ taxa tree all runs of both algorithms needed less than one second to compute a subtree with maximal *PD*. In our simulations, *gPDA* never consumed more than 8 s to achieve the subset of maximal phylogenetic diversity in the 1,000,000-taxa trees, whereas the longest run for the 1,000,000-taxa tree with *pPDA* amounts to 17 s. It should be noted that an implementation of the naïve version of the greedy algorithm (as derived from Steel, 2005) needs more than 30 min for $n = 10^5$ taxa (data not shown). In our simulations, *gPDA* is faster than *pPDA* if $k \leq 70\%$ of the taxa; otherwise, *pPDA* outperforms *gPDA*.

Typical applications do not deal with millions of taxa. But recently, Lewis and Lewis (2006) calculated *PD* for thousands of small trees of 150 taxa. We applied our algorithms to 10,000 trees generated from their data using MrBayes (Ronquist and Huelsenbeck, 2003). Both algorithms took less than 1.5 s to extract optimal *PD* subtrees for all generated trees. Hence *gPDA* and *pPDA* may serve as subroutines in such applications. In addition, this example resulted in a different discriminative point of $k = 40\%$ at which *pPDA* starts outperforming *gPDA*. Thus, the superiority of one algorithm over the other crucially depends on the tree shape.

Discussion

We have presented two versions of the greedy approach, *gPDA* and *pPDA*. They provide an efficient implementation to compute a subtree of given size k with maximal phylogenetic diversity. Thus, *gPDA* and *pPDA* may serve as convenient tools to compute subtrees for different sizes of k. The gain in speed is due to the trick that $\mathcal{S}$ does not contain all the interior nodes or taxa. Therefore both algorithms exhibit a worst-case performance less than $O(n \log n)$. Our simulations indicated that the tree shape influences the wall-clock computing time and the efficiency of the algorithms differently. A comprehensive study of all factors affecting computing time is beyond the scope of this study.

Steel (2005) proposed an extension of the *PD* score to accommodate the need to incorporate different measures of diversity, in which each taxon receives a weight depicting its estimated importance. This can be easily integrated into the algorithm by increasing the terminal branches with the weight of the corresponding taxa (Pardi and Goldman, 2005).

Pardi and Goldman also suggested another approach, namely to start with a user-defined initial set W. This permits the extension of W to a maximally divergent set starting with a non-optimal seed W. This application may be handy in comparative genomics where one has already some species sequenced and must decide which species to be sequenced next. We included this option in both algorithms.

Although the determination of one subset W of X with maximal *PD* is computationally efficient, it would be certainly worthwhile to explore the possibility of different sets $W_1, W_2, \ldots$ with the same maximal *PD*. The number of such sets can theoretically increase dramatically. In view of this combinatorial explosion, the question of how to measure phylogenetic diversity becomes important. For the algorithms, the precise nature of this measure is irrelevant as long as it can be mapped on the tree relating the taxa under consideration. Combining different measures of diversity may lead to more discriminative branch lengths and therefore reduce the hazard of multiple optimal sets.

In this context, confining the measure to genetic distances between the taxa may be helpful (Pardi and Goldman, 2005). However, different problems then arise. Presently, it is not at all clear how to adjust the algorithms for conflicting trees derived from the same set of taxa. It is well known that different regions of the genome provide trees with drastically different phylogenetic diversities due to violations of the molecular clock or due to varying rates of molecular evolution (Graur and Li, 2000). Sometimes trees derived from different regions may be different due to ancestral polymorphisms (Nei, 1987). The artificial example in Figure 3 illustrates the problem. For $k = 2$, we compute $W_1 = \{1, 3\}$, $W_2 = \{1, 4\}$ for trees $\mathcal{T}_1$ and $\mathcal{T}_2$, respectively. If we compute the pairwise distance between taxa as the sum of the pairwise distances in both trees, then the set $W_3 = \{3, 4\}$ displays the largest phylogenetic diversity. We also obtain W_3, if the tree is selected that provides the best least square fit to the distance sum (cf. Felsenstein, 2004). The crucial point is the fact that W_3 is neither maximal in T_1 nor in T_2. Thus, if we construct trees from different genomic regions and combine them naïvely, then the resulting tree and its derived optimal subtree with maximal diversity may not be the representative of the true underlying diversity. One way to address this would be to assign different weights to the different trees and then maximize the weighted average of the *PD*s calculated for different trees. However, more sophisticated algorithms may be required to cope adequately with genetic *PD*.

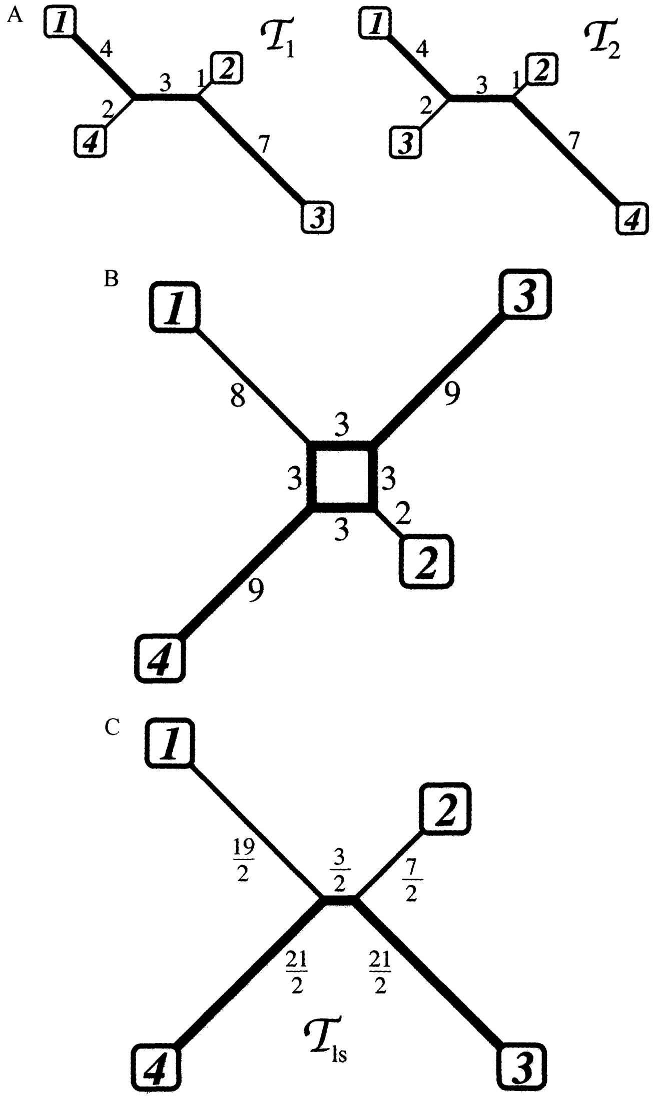

FIGURE 3. For the taxa 1, 2, 3, and 4, two different gene trees are observed that lead to two different PD_2 sets $\{1, 3\}$ and $\{1, 4\}$, respectively (A). In contrast, the resulting split graph generated by the sum of pairwise distances between taxa in $\mathcal{T}_1$ and $\mathcal{T}_2$ (B) and the least-squares, fit tree of the two gene trees (C) have the PD_2 set $\{3, 4\}$. Bold lines visualize the subgraphs formed by the respective PD_2 sets.

Acknowledgements

The authors would like to thank Dan Faith for helpful suggestions on the subject and Heiko Schmidt and Tanja Gesell for fruitful discussions. They also thank Mike Steel and anonymous referees for constructive comments. A computer program written in C++ is freely available from http://www.cibiv.at/software/pda/. Financial support from the Wiener Wissenschafts-, Forschungs, and Technologiefonds (WWTF) is greatly appreciated.

References

Cormen, T. H., C. E. Leiserson, R. L. Rivest, and C. Stein. 2001. Introduction to algorithms, 2 edition. MIT Press and McGraw-Hill, Cambridge, Massachusetts.

Faith, D. P. 1992. Conservation evaluation and phylogenetic diversity. Biol. Conserv. 61:1–10.

Felsenstein, J. 2004. Inferring phylogenies. Sinauer Associates, Sunderland, Massachusetts.

Graur, D., and W.-H. Li. 2000. Fundamentals of molecular evolution. 2nd edition. Sinauer Associates, Sunderland, Massachusetts.

Harding, E. F. 1971. The probabilities of rooted tree shapes generated by random bifurcation. Adv. Appl. Prob. 3:44–77.

Humphries, C. J., P. H. Williams, and R. I. Vane-Wright. 1995. Measuring biodiversity value for conservation. Annu. Rev. Ecol. Syst. 26:93–111.

Korte, B., L. Lovász, and R. Schrader. 1991. Greedoids. Algorithms and combinatorics. Springer Verlag, Berlin.

Lewis, L. A., and P. O. Lewis. 2006. Unearthing the molecular diversity of desert soil green algae. Syst. Biol. 54:936–947.

Nei, M. 1987. Molecular evolutionary genetics. Columbia University Press, New York.

Pardi, F., and N. Goldman. 2005. Species choice for comparative genomics: Being greedy works. PLoS Genet. 1:e71.

Ronquist, F., and J. P. Huelsenbeck. 2003. MrBayes 3: Bayesian phylogenetic inference under mixed models. Bioinformatics 19:1572–1574.

Steel, M. 2005. Phylogenetic diversity and the greedy algorithm. Syst. Biol. 54:527–529.

Williams, P. H., and M. B. Araujo. 2002. Apples, oranges and probabilities: Integrating multiple factors into biodiversity conservation with consistency. Environ. Model. Assess. 7:139–151.

First submitted 7 March 2006; reviews returned 21 April 2006;
final acceptance 15 June 2006
Associate Editor: Mike Steel