Syst. Biol. 56(3):431–444, 2007

ISSN: 1063-5157 print / 1076-836X online
DOI: 10.1080/10635150701411279

Resource-Aware Taxon Selection for Maximizing Phylogenetic Diversity

FABIO PARDI AND NICK GOLDMAN

EMBL—European Bioinformatics Institute, Wellcome Trust Genome Campus, Hinxton, Cambridge CB10 1SD, UK;
E-mail: pardi@ebi.ac.uk (F.P.)

Abstract.— Phylogenetic diversity (*PD*) is a useful metric for selecting taxa in a range of biological applications, for example, bioconservation and genomics, where the selection is usually constrained by the limited availability of resources. We formalize taxon selection as a conceptually simple optimization problem, aiming to maximize *PD* subject to resource constraints. This allows us to take into account the different amounts of resources required by the different taxa. Although this is a computationally difficult problem, we present a dynamic programming algorithm that solves it in pseudo-polynomial time. Our algorithm can also solve many instances of the Noah's Ark Problem, a more realistic formulation of taxon selection for biodiversity conservation that allows for taxon-specific extinction risks. These instances extend the set of problems for which solutions are available beyond previously known greedy-tractable cases. Finally, we discuss the relevance of our results to real-life scenarios. [Biodiversity conservation; comparative genomics; dynamic programming; phylogenetic diversity; Noah's Ark Problem; species choice; taxon selection.]

The *phylogenetic diversity* (*PD*) of a set of taxa (Faith, 1992), loosely defined as the total length of the evolutionary tree connecting them, is a measure of increasing importance in a number of areas in biology, in particular biodiversity conservation (Crozier, 1997; Barker, 2002; Mace et al., 2003; Forest et al., 2007) and comparative genomics (see Pardi and Goldman, 2005, and references therein). In bioconservation, the protection from extinction of species or populations that have a large total *PD* (or, equivalently, that represent a large portion of evolutionary history: Nee and May, 1997) is regarded as a good way to preserve genetic diversity (Crozier, 1997) and, more generally, diversity in the biological features of the existing organisms (Faith, 1992). In genomics, on the other hand, comparing homologous sequences with a high total divergence (which is to sequences what *PD* is to taxa) allows testing of evolutionary hypotheses and detection of various genomic features with high statistical power (Thomas et al., 2003; Eddy, 2005; Pardi and Goldman, 2005). Comparative genomics, like biological conservation, is therefore better done on phylogenetically diverse sets of genes (or individuals, populations or species: the unit is not important and we will use the word taxa throughout). Genome sequencing projects are increasingly aware of this (Margulies et al., 2005).

Often it will not be possible to save every existing species from large extinction events (such as the one we are currently causing) or to sequence every available genome. It is therefore natural to ask how choices should be made in these areas—"the agony of choice" is a popular phrase in bioconservation (Vane-Wright et al., 1991; Crozier, 1992). Because *PD* has come to be regarded as a good metric for measuring taxon importance, there has recently been interest in the formal optimization problems associated with it: how should we select taxa in order to maximize the resulting *PD*? The simplest of these problems, consisting of selecting a given number (decided a priori) of taxa, is easily solved by a greedy strategy (Steel, 2005; Pardi and Goldman, 2005). However, this scenario assumes that it is feasible to determine in advance the number of taxa that will be dealt with by the available resources, which is only true if the taxa require (roughly) the same amount of resources. In reality, this is usually not true: in bioconservation, for example, we may be designing a protected geographical region of fixed area, and different species will typically require different amounts of land; similarly, in the case of sequencing, different genomes have different sizes and therefore will have different costs, not only in terms of money but also of time and instruments required for sequencing. Potentially, a choice will have to be made between selecting few "expensive" taxa or many "cheap" ones.

Assuming that "costs" (whatever the available resource, e.g., money, time, labor, machinery, space, etc.) can be roughly quantified, a possible approach to limit "expenditure" could consist of modifying the evolutionary tree used as a basis for calculating *PD* by shortening each terminal branch by an amount depending on the cost of its taxon, so that the selection of costly taxa would be discouraged (Steel, 2005; Pardi and Goldman, 2005). This is not very satisfying; for example, there is no guarantee that the selected taxa will have maximum *PD* among all the other choices of taxa with the same total cost.

Here, we adopt a more direct approach: given taxon-associated costs and an estimate of the available resources, which we naturally call the *budget*, we aim to select the set of taxa of maximum *PD* among those sets with total cost at most equal to the budget. This can be re-expressed using some formalisms that will be useful throughout this paper. Let $\mathcal{X}$ be the chosen *phylogenetic scope* (Pardi and Goldman, 2005), i.e., the set of taxa we aim to select from, and $\mathcal{T}_{\mathcal{X}}$ their (possibly rooted) phylogenetic tree, where all branches have non-negative lengths. Each taxon (e.g., species or sequence) $s \in \mathcal{X}$ has a nonnegative integer cost c_s. We aim to

$$\begin{gathered}\text{find a subset } S \subseteq \mathcal{X} \text{ so as to} \\ \text{maximize } PD(S) \\ \text{subject to } \sum_{s \in S} c_s \leq B\end{gathered} \tag{1}$$

where B is an integer representing the budget.

Although $PD(S)$ has been simply defined as the total length of the smallest subtree of $\mathcal{T}_\mathcal{X}$ connecting all taxa in S (the "minimum spanning path" of Faith, 1992), there have been different interpretations of this definition in the literature (see, for example, the discussion in Faith and Baker, 2006; Crozier et al., 2006). These probably derive from different understandings of what a subtree is: if $\mathcal{T}_\mathcal{X}$ is a rooted tree, do its subtrees necessarily include its root? For example, does the smallest subtree of the mammals that contains human and chimp necessarily contain also the common ancestor of all mammals? It is clear that different ways of answering this question lead to different values of PD and sometimes to different solutions to the optimization problem (1). It is therefore useful to distinguish between the two possible definitions of PD. We will call the *unrooted phylogenetic diversity* (uPD) of a set of taxa $S \subseteq \mathcal{X}$ the total length of the smallest unrooted subtree of $\mathcal{T}_\mathcal{X}$ connecting all the taxa in S but not necessarily the root (assuming there is one). Conversely, when $\mathcal{T}_\mathcal{X}$ is rooted, the total length of the smallest subtree of $\mathcal{T}_\mathcal{X}$ having the same root as $\mathcal{T}_\mathcal{X}$ and connecting all the taxa in S will be referred to as the *rooted phylogenetic diversity* (rPD) of S (see Fig. 1). Clearly, denoting the root of $\mathcal{T}_\mathcal{X}$ by ρ (and assuming $\rho \in \mathcal{X}$), we have that $rPD(S) = uPD(S \cup \{\rho\})$. In practice, the choice between these two measures will be determined by the intended application: uPD seems more appropriate for comparative genomics (as most sequence comparison techniques are independent of root placement), whereas rPD seems to be preferred in conservation biology (Rodrigues and Gaston, 2002). Furthermore, as we will show in the following, the optimization problems associated with these two metrics can be of rather different difficulty, with uPD generally posing more problems than rPD. Insisting on this distinction is therefore not merely a pedantic exercise.

Incidentally, this ambiguity in the notion of PD is not surprising if we consider that Faith's paper introducing PD (Faith, 1992) was itself somewhat ambiguous: whereas one of the formal results (Faith, 1992: 4, last line) only holds for uPD, the only example that allows discrimination between the two interpretations (Faith, 1992: fig. 3a, set R3), is consistent with rPD but not uPD. In later papers (e.g., Moritz and Faith, 1998; Faith and Baker, 2006), Faith and colleagues clarified their preference for rPD, but the alternative definition for PD has become widespread (e.g., Crozier, 1997; Barker, 2002; Steel, 2005; Minh et al., 2006) and other authors even refer to rPD as *evolutionary history* (Nee and May, 1997).

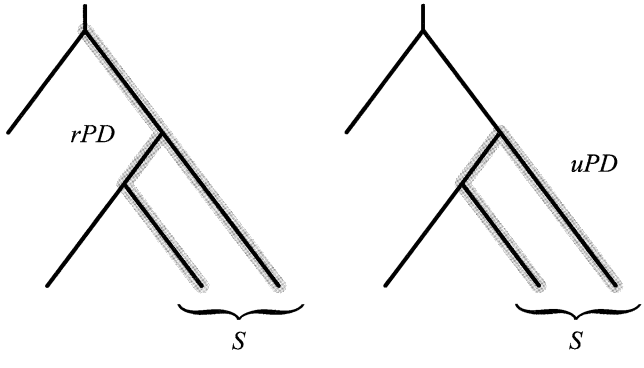


FIGURE 1. Difference between two definitions of PD. Highlighted are the branches whose lengths are summed in order to give $rPD(S)$ (left) and $uPD(S)$ (right). Assuming branch lengths are proportional to those in the figure, S is an optimal subset of size 2 for rPD but not uPD (uPD-optimal subsets of size 2 consist of the taxon on the left of the root, plus any other taxon). This shows that the solution of problem (1) is dependent on the definition of PD.

One way of solving problem (1) would be to calculate the cost and PD for all 2^n subsets of the taxa, where $n = |\mathcal{X}|$ is the number of taxa being selected among. However, this is not feasible even for moderate values of n, and therefore alternative approaches must be sought. The next two sections give novel algorithms that efficiently solve problem (1) for both definitions of PD. Before then, however, some considerations are needed. First, these algorithms will assume that $\mathcal{T}_\mathcal{X}$ is a bifurcating tree, which clearly is not a limitation, as every multifurcating tree can be resolved into an equivalent bifurcating tree in which the new (internal) branches have length 0. Second, note that the formulation (1) can also be used to express the scenario whereby a number of taxa have already been conserved (or sequenced): the search for a maximally diverse *extension* (Pardi and Goldman, 2005) of an initial set $I \subset \mathcal{X}$ can be implemented by solving problem (1) where all leaves in I have their cost set to 0. Third, note that all costs and the budget are (implicitly in the following) assumed to be integers. As will be shown, the algorithms' running times grow quadratically with the budget B. It is therefore important to express all costs and budget as multiples of large units to limit the value of B and, consequently, running times. This granularity is not unrealistic, since real-life resources are inherently discrete (e.g., currencies) and costs and budgets will often be known not with great precision but only approximately.

Compared to the problems for which an efficient solution is known (Steel, 2005; Pardi and Goldman, 2005), problem (1) is clearly a step towards realism. In the context of biodiversity conservation, however, there is an important factor that is not taken into account: different taxa have different risks of extinction (Witting and Loeschcke, 1993, 1995). For example, it is clear that if a species is not endangered at all, then expending resources to conserve this or a very closely related species is not a good choice. It is intuitive that conservation efforts should concentrate on the most endangered species, although under some particular conditions the contrary seems to be true (Weitzman, 1998). These aspects have been formalized by Weitzman (1992, 1993, 1998) into the suggestively named "Noah's Ark Problem" (NAP) (Weitzman, 1998), which is receiving increasing attention especially in environmental and ecological economics (Simianer et al., 2003; Reist-Marti et al., 2003; van der Heide et al., 2005). Although Weitzman (1998) derived a myopic, or greedy, ranking criterion that assigns each taxon a conservation priority score, an exact algorithmic solution to the NAP remains an open problem.

Hartmann and Steel (2006) have shown that some special cases of the NAP can be solved, again, with a greedy

algorithm. The algorithms we present here allow the resolution of further relatively general cases of the NAP, extending the scenarios considered by Hartmann and Steel. Although the main focus of this paper is on results that can be used in more than just biodiversity conservation, we will illustrate the consequences of our work for the NAP in Applications to the Noah's Ark Problem.

Finally, we note that problem (1) (and the NAP we define below, which generalizes it) is NP-hard, as the knapsack problem, another well-known NP-hard problem (Cormen et al., 2001), is simply its special case for star-shaped trees (Hartmann and Steel, 2006). Technically, our algorithms are examples of pseudo-polynomial time algorithms: although they run in time polynomial in B, this number may itself grow exponentially in the size of the input (see Garey and Johnson, 1979, for further discussion).

A Dynamic Programming Algorithm for the Rooted Case

When phylogenetic diversity is defined as rPD (as we implicitly assume throughout this section), problem (1) can be solved with a dynamic programming algorithm, which we will describe in this section. In the next section we will show that this algorithm can be extended to solve the unrooted (uPD) version of problem (1).

The key observation that allows the use of a dynamic programming algorithm is that optimal solutions to problem (1) can be simply decomposed into optimal solutions to its subproblems—technically, problem (1) is said to have *optimal substructure* (Cormen et al., 2001). As a consequence, optimal solutions to problem (1) can be constructed by first tackling and solving its subproblems and then combining the optimal solutions thus found.

In order to see what this means in practice, we need to introduce some concepts that allow us to apply problem (1) to smaller portions of $\mathcal{T}_\mathcal{X}$. We define a *clade* as any subtree of $\mathcal{T}_\mathcal{X}$ consisting of a branch a and everything else below a in $\mathcal{T}_\mathcal{X}$ (note that we imagine $\mathcal{T}_\mathcal{X}$ with its root at the top); a clade should be thought of as rooted at the top of its top branch a. In a bifurcating tree, every clade is composed of a root branch and, possibly, by two other clades, which we call its *subclades*. (For example, in Fig. 2, left, clade $\mathcal{T}$ has two subclades, $\mathcal{L}$ and $\mathcal{R}$, whereas $\mathcal{L}$ is composed of one terminal branch and no subclades.)

A way to decompose problem (1) into smaller subproblems is to ask what is the best way to invest a given part of the budget into a given clade (i.e., which of the taxa in this clade should be selected given that part of the budget). It turns out that we can answer this question for all clades $\mathcal{T}$ and sub-budgets $b \leq B$ incrementally, starting from the clades consisting of only a terminal branch and then using the solutions already found to construct the solutions for larger clades.

Formally, the subproblems consist of finding, for every $b \in \{0, 1, \ldots, B\}$ and every clade $\mathcal{T}$, a subset S of the taxa in $\mathcal{T}$ that maximizes $rPD_\mathcal{T}(S)$ subject to $\sum_{s\in S} c_s \leq b$. Here $rPD_\mathcal{T}(S)$ denotes the rooted PD calculated as if $\mathcal{T}$ was the whole tree (e.g., in Fig. 2, $rPD_\mathcal{R}(\{E\}) = 3.0$). S will be called an (*optimal*) *solution* for $\mathcal{T}$ and b. Note that one of these subproblems (the one with $\mathcal{T} = \mathcal{T}_\mathcal{X}$ and $b = B$) simply coincides with problem (1) itself ($\mathcal{T}_\mathcal{X}$ can always be seen as having a root branch, possibly of zero length, and therefore is a clade itself). Optimal solutions to these subproblems (or more precisely some sufficient information about them, as we will see later) can be stored in a table (the *solutions table*) whose rows correspond to all the different clades and whose columns correspond to all the sub-budgets 0, 1, ..., B (see Fig. 2). Clearly, position $(\mathcal{T}, b)$ will contain (information about) an optimal solution for $\mathcal{T}$ and b. We will show that this table can be completed one row at a time, starting from the bottom and going up.

First, the solutions for the clades consisting of only a terminal branch (leading to, say, taxon s) are simply either the empty set $\emptyset$ or $\{s\}$, depending on whether the taxon is too expensive to be taken with the available sub-budget (i.e., $c_s > b$), or not ($c_s \leq b$), respectively. Therefore the rows of the solutions table corresponding to terminal branches (in Fig. 2, the 3rd, 4th, 6th, and the last two) can be filled without looking at the content of any other row.

Second, when instead a clade $\mathcal{T}$ contains two subclades ($\mathcal{L}$ and $\mathcal{R}$), an optimal solution S for $\mathcal{T}$ and sub-budget b will simply amount to the union $S = S_\mathcal{L} \cup S_\mathcal{R}$ of two optimal solutions for two other subproblems: $S_\mathcal{L}$ will be optimal for $\mathcal{L}$ and some sub-budget $i \leq b$, whereas $S_\mathcal{R}$ will be optimal for $\mathcal{R}$ and $b - i$. (The reason for this is simple: S is naturally partitioned into $S_\mathcal{L}$ and $S_\mathcal{R}$, containing the taxa in $\mathcal{L}$ and $\mathcal{R}$, respectively; calling i the total cost of $S_\mathcal{L}$, if either $S_\mathcal{L}$ were not optimal for $\mathcal{L}$ and i or $S_\mathcal{R}$ were not optimal for $\mathcal{R}$ and $b - i$, then we could replace this suboptimal choice of taxa with a better one and therefore improve also S, but this would contradict the fact that S is optimal.) For example, in Figure 2, an optimal solution for $\mathcal{T}$ and $b = 4$ is {C, E}, where {C} is optimal for $\mathcal{L}$ and sub-budget 2, and {E} is optimal for $\mathcal{R}$ and 2.

Importantly, if we have already calculated and stored optimal solutions for $\mathcal{L}$ and $\mathcal{R}$ and for all sub-budgets, it becomes possible to find a solution for $\mathcal{T}$ and any given b: denoting by $S_\mathcal{L}^{(i)}$ the solution stored for $\mathcal{L}$ and i (and similarly for $S_\mathcal{R}^{(j)}$), just compare all the subsets $S_\mathcal{L}^{(0)} \cup S_\mathcal{R}^{(b)}$, $S_\mathcal{L}^{(1)} \cup S_\mathcal{R}^{(b-1)}$, ..., $S_\mathcal{L}^{(b)} \cup S_\mathcal{R}^{(0)}$ and take the one with the largest rPD. This is guaranteed to find an optimal solution for $\mathcal{T}$ and b, because the possibility of decomposing an optimal solution S into $S_\mathcal{L} \cup S_\mathcal{R}$ (where $S_\mathcal{L}$ is optimal for $\mathcal{L}$ and some $i \in \{0, 1, \ldots, b\}$, and $S_\mathcal{R}$ is optimal for $\mathcal{R}$ and $b - i$) implies that $rPD(S) = rPD(S_\mathcal{L} \cup S_\mathcal{R}) = rPD(S_\mathcal{L}^{(i)} \cup S_\mathcal{R}^{(b-i)})$ and therefore $S_\mathcal{L}^{(i)} \cup S_\mathcal{R}^{(b-i)}$ is also optimal. Note that if there are multiple optimal solutions for $\mathcal{L}$ and i (or for $\mathcal{R}$ and $b - i$), some of which are empty and some of which are not (possible if there are paths of zero length from the root of the clade to some of the taxa), we must ensure that the stored solution $S_\mathcal{L}^{(i)}$ (or $S_\mathcal{R}^{(b-i)}$) is nonempty. Otherwise, we may have that the union of optimal solutions

is nonoptimal (specifically, when $rPD(S_{\mathcal{L}} \cup S_{\mathcal{R}}) > 0$ and $rPD(S_{\mathcal{L}}^{(i)} \cup S_{\mathcal{R}}^{(b-i)}) = rPD(\emptyset \cup \emptyset) = 0$).

Consequently, we can fill the entire solutions table one row at a time: because the content of the row for a clade $\mathcal{T}$ can be completely derived from the content of the rows for its subclades $\mathcal{L}$ and $\mathcal{R}$, we just need to make sure that, whenever we fill the row for a clade, the rows for its subclades have already been filled. This can be achieved by dealing with the clades in a bottom-up order: let the rows in the table be ordered according to a top-down traversal of all clades (as illustrated in Fig. 2); then, fill the rows from the last to the first. Once the entire table has been filled, the solution to problem (1) is available from its top right corner.

Although storing entire solutions in the solutions table is feasible, this is certainly not efficient, as solutions can contain many taxa and therefore require a variable amount of memory and time to be handled. As we will show, instead of storing solutions, it is sufficient to retain their rPD and the way the sub-budget should be partitioned between the two subclades (if there are any). More precisely, let $S_{\mathcal{T}}^{(b)}$ be the solution found for clade $\mathcal{T}$ and sub-budget b; instead of storing $S_{\mathcal{T}}^{(b)}$, we will store two quantities: (a) the maximum rPD achievable in $\mathcal{T}$ with sub-budget b, which we call $\lambda_{\mathcal{T}}(b)$ and is equal to $rPD_{\mathcal{T}}(S_{\mathcal{T}}^{(b)})$; and (b) (only when $\mathcal{T}$ has two subclades) the sub-budget that solution $S_{\mathcal{T}}^{(b)}$ allocates to $\mathcal{T}$'s left subclade, which we call $\iota_{\mathcal{T}}(b)$—obviously this also determines the sub-budget to allocate to the right subclade, $b - \iota_{\mathcal{T}}(b)$. Note that this does not mean that the actual expenditures in the subclades need equal $\iota_{\mathcal{T}}(b)$ and $b - \iota_{\mathcal{T}}(b)$, but rather that the left part of $S_{\mathcal{T}}^{(b)}$ is optimal for $\iota_{\mathcal{T}}(b)$, and its right part for $b - \iota_{\mathcal{T}}(b)$. In Figure 2, each cell in the solutions table shows $\lambda_{\mathcal{T}}(b)$ at the top and, when appropriate, in the two bottom corners, $\iota_{\mathcal{T}}(b)$ on the left and (for illustrative purposes) $b - \iota_{\mathcal{T}}(b)$ on the right, which indicates the way the sub-budget should be partitioned between the two subclades.

Note that it is precisely thanks to storing $\iota_{\mathcal{T}}(b)$ that solutions need not be memorized; this information (once available) allows us to reconstruct the optimal solution found for any given $\mathcal{T}$ and b: just visit all the clades below $\mathcal{T}$ in a top-down fashion always using the $\iota_{\mathcal{T}}$ values to find out what part of b should be devoted to each clade; in the end, b will have been broken into all the expenditures to allocate to each taxon in $\mathcal{T}$; a taxon s should be included in the solution only if the expenditure for s is greater or equal to its cost c_s (see Appendix, Algorithm 3, RECONSTRUCT($\mathcal{T}$, b), for a formalization of this procedure). For example, imagine that we wish to reconstruct the solution for clade $\mathcal{T}$ and sub-budget 4 in Figure 2; the two values at the bottom corners of the table entry for $\mathcal{T}$ and 4 (shaded gray) indicate that we should allocate 2 to $\mathcal{L}$ and 2 to $\mathcal{R}$. $\mathcal{L}$ only contains taxon C, whose cost is exactly 2 and therefore should be selected. $\mathcal{R}$ has two subclades, one containing D and the other E; the table entry for $\mathcal{R}$ and 2 indicates that nothing should be spent in the left subclade (so D should not be selected, as its cost is greater than 0) and that 2 should be assigned to the right subclade (so we do select E, as its cost is exactly 2); therefore, the solution for $\mathcal{T}$ and 4 is {C, E}.

The solutions table is filled with $\lambda_{\mathcal{T}}(b)$ and $\iota_{\mathcal{T}}(b)$ values in a way analogous to the one we described above in terms of whole solutions. The clades are visited in a bottom-up order. For the clades only consisting of a terminal branch (leading to, say, taxon s), the $\lambda_{\mathcal{T}}(b)$ values are set to 0 for entries with b up to (but not including) c_s and to the length of the terminal branch for the remaining entries; the $\iota_{\mathcal{T}}(b)$ are left undefined, as they have no meaning for these clades. For example, see the solutions table row for $\mathcal{L}$ (shaded red) in Figure 2.

When instead we visit a clade $\mathcal{T}$ composed of a branch (a) and two subclades ($\mathcal{L}$ and $\mathcal{R}$), we need to consider two cases. First, for all the entries with b smaller than the minimum cost among the taxa in $\mathcal{T}$ (which we denote by $\check{c}_{\mathcal{T}}$), $\lambda_{\mathcal{T}}(b)$ is set to 0, as clearly the sub-budget b is not enough to cover the cost of any of the taxa in $\mathcal{T}$; note also that for these entries $\iota_{\mathcal{T}}(b)$ can be set to any $i = 0, 1, \ldots, b$, as any of these values will lead to reconstructing the empty solution (for example, see the first two entries in the table row for $\mathcal{T}$ in Fig. 2). Second, for the remaining entries (those with $b \geq \check{c}_{\mathcal{T}}$), $\lambda_{\mathcal{T}}(b)$ is set to the maximum value among $t_a + \lambda_{\mathcal{L}}(0) + \lambda_{\mathcal{R}}(b)$, $t_a + \lambda_{\mathcal{L}}(1) + \lambda_{\mathcal{R}}(b-1), \ldots, t_a + \lambda_{\mathcal{L}}(b) + \lambda_{\mathcal{R}}(0)$ (where t_a represents the length of a), as this is the rPD of the best possible combination of complementary solutions for $\mathcal{L}$ and $\mathcal{R}$; the corresponding $\iota_{\mathcal{T}}(b)$ is set to a value of $i \in \{0, 1, \ldots, b\}$ that maximizes $\lambda_{\mathcal{L}}(i) + \lambda_{\mathcal{R}}(b-i)$. For example, suppose we aim to fill the entry for $\mathcal{T}$ and 4 in Figure 2. Because we are proceeding in a bottom-up fashion, the rows corresponding to $\mathcal{L}$ and $\mathcal{R}$ have already been filled, and the $\lambda_{\mathcal{R}}$ and $\lambda_{\mathcal{L}}$ values are all available. Therefore, $\lambda_{\mathcal{T}}(4) = \max\{1.0 + \lambda_{\mathcal{L}}(0) + \lambda_{\mathcal{R}}(4), 1.0 + \lambda_{\mathcal{L}}(1) + \lambda_{\mathcal{R}}(3), \ldots, 1.0 + \lambda_{\mathcal{L}}(4) + \lambda_{\mathcal{R}}(0)\} = \max\{5.0, 5.0, 5.0, 2.0, 2.0\} = 5.0$. This corresponds to consideration of combining the first entry shaded in red with the fifth in blue, the second in red with the fourth in blue, and so on; in the end we check which of these combinations has given the largest rPD. The value of $\iota_{\mathcal{T}}(4)$ is set accordingly: in this case, there are three equivalent combinations and $\iota_{\mathcal{T}}(4)$ could be set to 0, 1, or 2 (leading to two different but equally good solutions).

In summary, the λ and ι values are calculated with the following recursions (which assume that $\mathcal{T}$ is composed of branch a and, possibly, subclades $\mathcal{L}$ and $\mathcal{R}$):

$$\lambda_{\mathcal{T}}(b) = \begin{cases} 0 & \text{if } b < \check{c}_{\mathcal{T}}, \\ t_a & \text{if } a \text{ is terminal and } b \geq \check{c}_{\mathcal{T}}, \\ t_a + \max\limits_{i \in \{0,\ldots,b\}} \{\lambda_{\mathcal{L}}(i) + \lambda_{\mathcal{R}}(b-i)\} & \text{otherwise.} \end{cases}$$

$$\iota_{\mathcal{T}}(b) = \begin{cases} \text{undefined} & \text{if } \mathcal{T} \text{ has no subclades,} \\ \arg\max\limits_{i \in \{0,\ldots,b\}} \{\lambda_{\mathcal{L}}(i) + \lambda_{\mathcal{R}}(b-i)\} & \text{otherwise.} \end{cases}$$

The above arg max term indicates the value of i that maximizes the expression on its right; when there are several such values, it indicates any one of them (e.g., chosen randomly) or, in the special case where $b \geq \check{c}_{\mathcal{T}}$ and $\lambda_{\mathcal{L}}(i) + \lambda_{\mathcal{R}}(b - i) = 0$ for all i, any value of i such that the resulting $\iota_{\mathcal{T}}(b)$ will cause reconstruction of a nonempty solution (this can be achieved by taking either $i = \check{c}_{\mathcal{L}}$ if $\check{c}_{\mathcal{L}} \leq b$, or $i = b - \check{c}_{\mathcal{R}}$ if $\check{c}_{\mathcal{R}} \leq b$). Implicitly storing the empty solution would either be wrong (if $t_a > 0$, any optimal solution is certainly nonempty) or (if $t_a = 0$) might lead to constructing a nonoptimal solution further up in the tree (see discussion above).

The various minimum costs $\check{c}_{\mathcal{T}}$ for all clades should also be derived and stored. For a given $\mathcal{T}$, this can be done when we set about filling its row, by either simply copying c_s (if $\mathcal{T}$ is a terminal branch leading to s) or taking the minimum between $\check{c}_{\mathcal{L}}$ and $\check{c}_{\mathcal{R}}$ (if $\mathcal{T}$ contains subclades $\mathcal{L}$ and $\mathcal{R}$). In Figure 2, the $\check{c}_{\mathcal{T}}$ values are reported on the right of the solutions table.

Once all the λ and ι values have been derived in the solutions table, the solution implicitly found for the top right entry of the table is also a solution to problem (1) and it can be reconstructed using the $\iota_{\mathcal{T}}$ values in the way we described above. However, there may be more than one optimal solution to problem (1), all equally good with respect to *rPD* but possibly involving different overall expenditures. It is not guaranteed that the solution reconstructed as described above is the cheapest among them. Although this is not required by the problem formulation (1), this is clearly a desirable property and easy to achieve: by looking at the solutions found for smaller sub-budgets, we can check if the budget can be reduced without affecting the optimal *rPD*. This corresponds to scanning the solutions table from its top right entry towards the left, until a reduction in $\lambda_{\mathcal{T}_{\mathcal{X}}}(b)$ is observed. The solution for the last (i.e., least) b with $\lambda_{\mathcal{T}_{\mathcal{X}}}(b) = \lambda_{\mathcal{T}_{\mathcal{X}}}(B)$ is a minimal-cost *rPD*-optimal solution and can be reconstructed in the usual way. In some cases it may even be of interest to derive all (minimal-cost) optimal solutions, which could be achieved by storing mutiple $\iota_{\mathcal{T}}(b)$ values and using all of them in the reconstruction at the end. However, we note that the number of solutions to reconstruct may grow exponentially in the size of the problem.

This concludes the description of our dynamic programming algorithm for maximizing *rPD* subject to cost constraints. A different description, less verbose and directly convertible into computer code, is given by the pseudocode in Algorithms 1, 2, and 3 in Appendix. Additionally, the dynamic programming algorithm could be modified to solve problem (1) for trees with branches of any length (i.e., where negative lengths are allowed), but for simplicity we do not describe this here.

Regarding the computational complexity of our algorithm, the calculation of a single entry in the solutions table requires $O(b) = O(B)$ time, as all possible ways to split sub-budget b may need to be examined. This must be repeated for each of the $(2n - 1)(B + 1) = O(nB)$ subproblems (where $n = |\mathcal{X}|$), giving a total of $O(nB^2)$ operations for filling the solutions table. The reconstruction of an optimal solution for problem (1) from the top right entry only takes $O(n)$ time, as it consists in a top-down traversal of all the $2n - 1 = O(n)$ clades of $\mathcal{T}_{\mathcal{X}}$, in which each clade can be dealt with in constant time. Therefore, the entire algorithm has time complexity $O(nB^2)$. Memory complexity is dominated by the size of the solutions table, and so is $O(nB)$.

Finally, we note that problem (1) could also be formulated as an integer linear programming (ILP) problem (in a way analogous to that of Rodrigues and Gaston, 2002) and solved with standard off-the-shelf techniques. However, there are no guarantees that the running time of these algorithms would be better than exponential in n.

The Unrooted Case

We now turn to solving problem (1) with *PD* defined as *uPD*, which we call the *unrooted problem*. An example is given in Figure 3. At first glance, an obstacle to its solution seems to be that there is no evident way to break it into smaller problems: even if we solve the unrooted problem on portions of $\mathcal{T}_{\mathcal{X}}$, this does not tell us much about the solution for the whole tree.

The key observation here is that a solution to the unrooted problem, if not the empty set, is equal to $\{s\} \cup R$, where R is a solution to the rooted problem with budget $B - c_s$ applied to $\mathcal{T}^s_{\mathcal{X}}$, which denotes the version of $\mathcal{T}_{\mathcal{X}}$ rooted in taxon s. (For example, in Figure 3, {A,C,D,E}, the optimal solution to the unrooted problem, can be written as {A} ∪ {C,D,E}, where {C,D,E} is a solution to the rooted problem on $\mathcal{T}^{\mathrm{A}}_{\mathcal{X}}$ with budget $B - c_{\mathrm{A}} = 8 - 1 = 7$.)

This allows us to reduce the unrooted problem to a number of related rooted problems: we could iteratively root $\mathcal{T}_{\mathcal{X}}$ in each of its taxa s and calculate a solution R_s to the rooted problem with budget $B - c_s$ (or skip s if $c_s > B$); any of the subsets $\{s\} \cup R_s$ with the largest *uPD* (or, equivalently, any of the ones with the largest $rPD(R_s)$) is a solution to the unrooted problem.

However, this procedure would involve repeating the *rPD*-maximization algorithm once for every taxon, thus requiring $O(n^2B^2)$ time. A more efficient approach, which we describe in the Appendix, consists of extending our notion of clade so that it includes all the different taxon-rootings $\mathcal{T}^s_{\mathcal{X}}$; as before, we solve the rooted subproblems (with many possible sub-budgets) for all the clades in $\mathcal{T}_{\mathcal{X}}$. Because this now includes additional clades not present before, we devise a new ordering of the clades so that we can incrementally derive new solutions from the previously calculated ones. In the end, we compare the rooted solutions found for the various $\mathcal{T}^s_{\mathcal{X}}$; any of the best ones will provide us with an optimal solution to the unrooted problem.

This more efficient algorithm for the unrooted problem is practically equivalent in computational complexity to the one in the last section. As described in the Appendix, there are now $2(2n - 3)$ clades, and therefore still $O(n)$ rows in the solutions table (and $O(B)$ columns). Derivation of the solution for $\mathcal{T}$ and b from the other stored solutions and reconstruction of any of these solutions

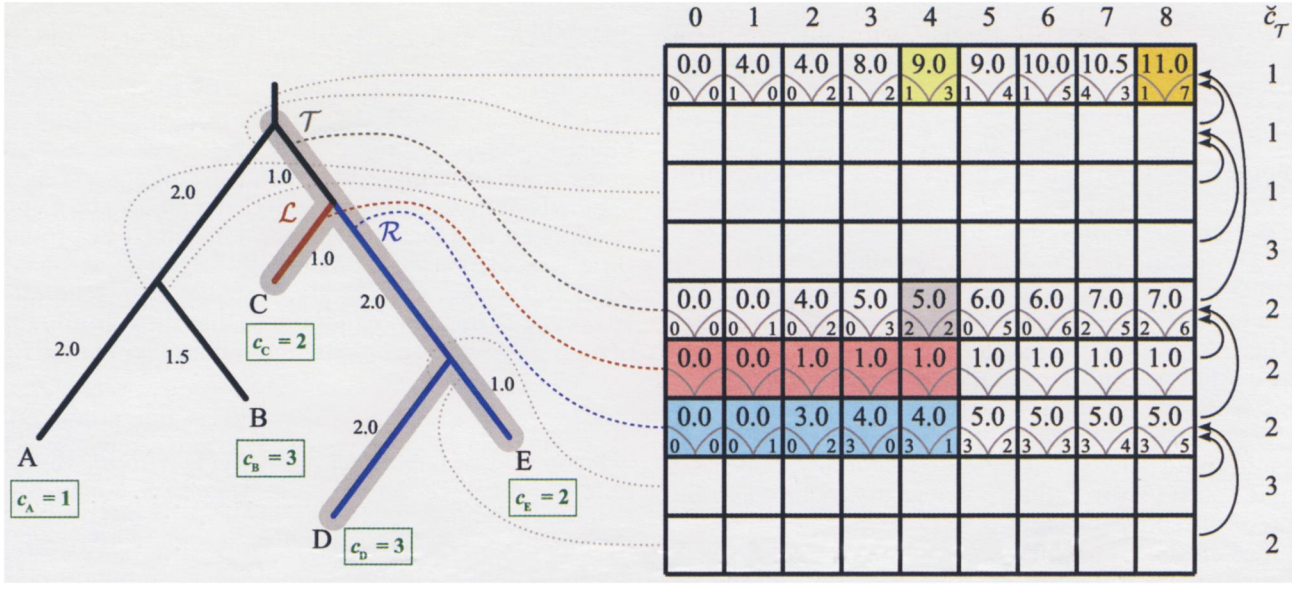


FIGURE 2. On the left, an instance of problem (1) (rooted case). Branch lengths are indicated by the numbers to the side of the branches. Taxon costs are in the boxes next to the leaves (green) and the budget is $B = 8$. The following clades of $\mathcal{T}_{\mathcal{X}}$ are highlighted: $\mathcal{T}$ (broad grey branches), $\mathcal{L}$ (red branches), $\mathcal{R}$ (blue branches). On the right, the corresponding solutions table (the content of some rows is omitted for clarity). The correspondence between clades and rows is indicated by dotted lines (colored in the cases of $\mathcal{T}$, $\mathcal{L}$, and $\mathcal{R}$). Rows are ordered according to a top-down visit of all clades of $\mathcal{T}_{\mathcal{X}}$: 1: ((A,B),(C,(D,E))); 2: (A,B); 3: A; 4: B; 5: (C,(D,E)); 6: C; 7: (D,E); 8: D; 9: E. To the right of the table are $\check{c}_{\mathcal{T}}$ values and arrows indicating the dependencies among the rows. Column headings 0 – 8 indicate sub-budgets b. The top right corner (shaded orange) of the solutions table indicates the optimal *rPD* (11, achieved by selecting taxa {A, C, D, E}). A naïve greedy algorithm—consisting of always selecting the taxon that adds most *PD* among the ones that can be selected with the currently available budget—would not work in this instance: after selecting D and A (at a cost of $3 + 1 = 4$), the remainder of the budget ($B - 4 = 4$) would be used on B, leading to a total *rPD* of 10.5. Also note that another greedy algorithm—consisting of always selecting the taxon with the highest ratio between added *rPD* and cost $c_\cdot$ (Hartmann and Steel, 2006)—would work in this instance but fails on the problem with budget $B = 4$; whereas the (unique) optimal solution for this budget is {A, D}, with an *rPD* of 9 (yellow cell), this greedy algorithm would initially select A and then E, thus including a taxon that is not part of the optimal solution.

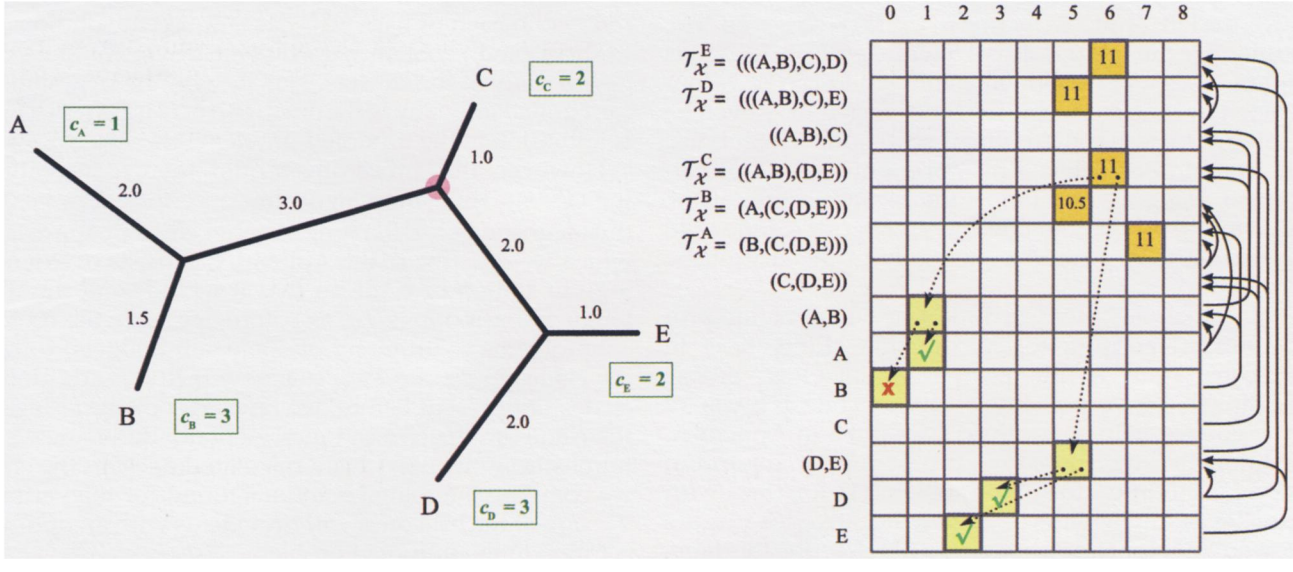


FIGURE 3. On the left, an instance of problem (1) (unrooted case). Branch lengths are indicated by the numbers to the side of the branches. Taxon costs are in the boxes next to the leaves (green) and the budget is $B = 8$. On the right, the corresponding solutions table (partially filled, for clarity). Rows correspond to the clades specified to the left, whereas column headings 0–8 indicate sub-budgets b. Arrows to the right of the table indicate the dependencies amongst the rows; see Appendix for an explanation of the row ordering (and the use of the node highlighted in pink). Orange cells correspond to solutions to the rooted subproblem for clades $\mathcal{T}^{\circ}_{\mathcal{X}}$ and sub-budgets $B - c_\cdot$. Dotted arrows and yellow cells show the reconstruction of the solution {A, C, D, E} to the unrooted problem (visited taxa are marked with a green tick or a red cross, depending on whether they should be selected or not, respectively).

using the $\iota_{\mathcal{T}}$ values are achieved in exactly the same way as before and therefore still take $O(B)$ and $O(n)$ time, respectively. Comparing $rPD(R_s)$ for all the rooted solutions R_s in order to find the unrooted solution $\{s\} \cup R_s$ also takes just $O(n)$ time. Therefore, this algorithm has time complexity $O(nB^2)$ and memory complexity $O(nB)$.

Applications to the Noah's Ark Problem

In the NAP (Weitzman, 1998) each taxon $s \in \mathcal{X}$ in a phylogenetic tree $\mathcal{T}_{\mathcal{X}}$ survives until some future time with a probability p_s that depends on the expenditure that is put into conserving s. The objective is to subdivide a given budget B among the available taxa to maximize the rPD of the surviving taxa or, more precisely, the expected value of this random variable. By increasing the expenditure γ_s for taxon s, the probability of survival will also (generally) increase as a function $p_s(\gamma_s)$. In the original formulation (Weitzman, 1998), p_s grows linearly from an initial value a_s to a maximum probability b_s, as the expenditure is increased from 0 to a cost c_s. For this shape of $p_s(\gamma_s)$ (and others) optimal solutions to the NAP are "extreme"; i.e., for every taxon s (with the possible exception of one taxon), either nothing or the whole c_s is spent on it (Weitzman, 1998). Following this, Hartmann and Steel (2006) have cast the NAP in a form (not entirely equivalent to the original NAP) analogous to that of problem (1):

$$\begin{aligned} &\text{find a subset } S \subseteq \mathcal{X} \text{ so as to} \\ &\text{maximize } \mathbb{E}(rPD \mid S) \\ &\text{subject to } \sum_{s \in S} c_s \leq B. \end{aligned} \tag{2}$$

Here, $\mathbb{E}(rPD \mid S)$ denotes the expected rPD of the taxa that "survive," where taxa survive independently with a probability of either b_s or a_s, depending on whether $s \in S$ or not. It is easy to see that

$$\mathbb{E}(rPD \mid S) = \sum_a t_a \left(1 - \prod_{s \in C_a - S} (1 - a_s) \prod_{s \in C_a \cap S} (1 - b_s)\right) \tag{3}$$

(Hartmann and Steel, 2006), where the sum is over all branches of $\mathcal{T}_{\mathcal{X}}$ and C_a denotes the set of taxa below branch a (again, $\mathcal{T}_{\mathcal{X}}$ is pictured with its root at the top).

Note that problem (1) is the particular case of problem (2) obtained by setting $a_s = 0$ and $b_s = 1$ for all taxa. We will refer to problem (1) as the $0 \xrightarrow{c_s} 1$ NAP and to (2) as the $a_s \xrightarrow{c_s} b_s$ NAP. This notation indicates the probabilities of survival without and with conservation (left and right of the arrow) and the conservation costs (above); these may be constants or taxon-dependent (indicated by variables indexed by taxon s).

There is in fact a hierarchy of subproblems of the $a_s \xrightarrow{c_s} b_s$ NAP. The simplest of them is the $0 \xrightarrow{1} 1$ NAP, which can be solved with a greedy algorithm (Steel, 2005; Pardi and Goldman, 2005). There are other greedy-tractable regions in this hierarchy, notably the $0 \xrightarrow{c_s} 1$ NAP applied to an ultrametric tree and the $1 - q_s \xrightarrow{c} 1 - \kappa q_s$ NAP, where for every taxon s the initial probability of extinction q_s can be reduced by a constant factor κ ($0 \leq \kappa \leq 1$) by paying a constant price c (Hartmann and Steel, 2006).

We will show that other (larger) regions of this hierarchy can be solved with dynamic programming algorithms: our base algorithm for problem (1) already demonstrated this for the $0 \xrightarrow{c_s} 1$ NAP, and we now show that the same holds, more generally, for the $a_s \xrightarrow{c_s} 1$ NAP. This is the relatively realistic scenario whereby conservation projects with variable costs c_s completely ensure survival of the species to which they are applied, which can have different initial risks of extinction (a_s depends on s). This may be the case when a taxon can be saved by simply saving a few of its individuals, be it in a zoo or in a prophet's ark. An example of $a_s \xrightarrow{c_s} 1$ NAP is given in Figure 4: panel (a) gives a tree $\mathcal{T}_{\mathcal{X}}$ for 52 Madagascar lemurs (including all those in the current IUCN Red List of Threatened Species, IUCN, 2006) and putative costs c_s of conserving each of them, and panel (b) shows the probabilities of survival without conservation, a_s, and their effect on this problem, as explained below.

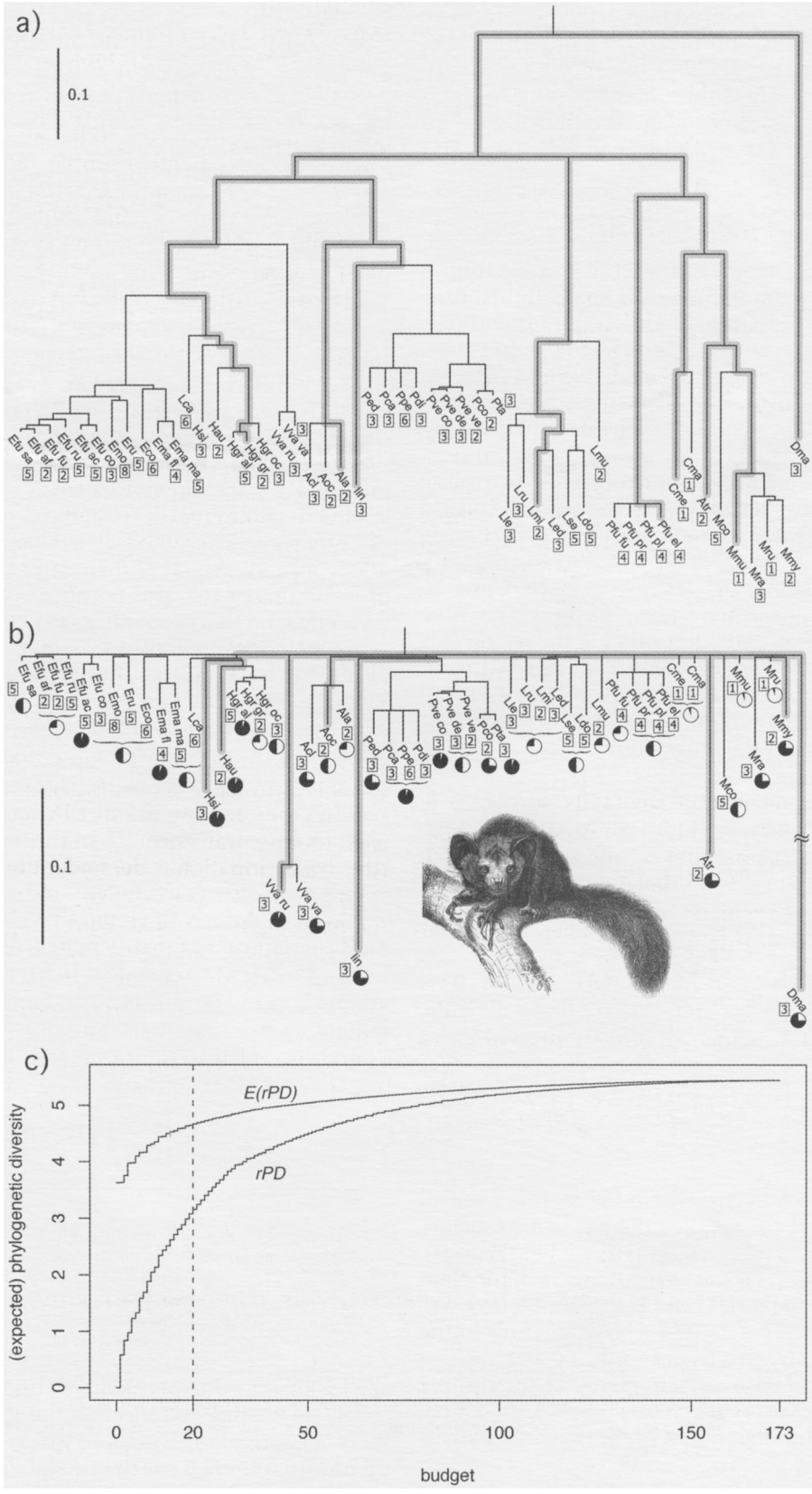
The key observation here is that, for any instance of the $a_s \xrightarrow{c_s} 1$ NAP, it is possible to transform the input tree $\mathcal{T}_{\mathcal{X}}$ into a new tree $\mathcal{T}'_{\mathcal{X}}$ so that the gain in $\mathbb{E}(rPD)$ obtained by conserving the taxa in S coincides with the rPD of S in $\mathcal{T}'_{\mathcal{X}}$, for any possible subset $S \subseteq \mathcal{X}$. As a consequence, we can solve any $a_s \xrightarrow{c_s} 1$ NAP by solving with our base algorithm the corresponding rPD-maximization problem on the transformed tree $\mathcal{T}'_{\mathcal{X}}$. In Figure 4, for example, the tree in panel (b) is the transformation $\mathcal{T}'_{\mathcal{X}}$ of the tree $\mathcal{T}_{\mathcal{X}}$ in panel (a).

Formally, define $\mathcal{T}'_{\mathcal{X}}$ as the tree obtained from $\mathcal{T}_{\mathcal{X}}$ by multiplying each branch length t_a by a factor equal to $\prod_{s \in C_a}(1 - a_s)$. Then, the gain in $\mathbb{E}(rPD)$ due to conserving the taxa in any subset S can be simply derived from Equation (3):

$$\begin{aligned} &\mathbb{E}(rPD_{\mathcal{T}_{\mathcal{X}}} \mid S) - \mathbb{E}(rPD_{\mathcal{T}_{\mathcal{X}}} \mid \emptyset) \\ &\quad = \sum_{a:\, C_a \cap S = \emptyset} t_a \left[1 - \prod_{s \in C_a} (1 - a_s)\right] \\ &\qquad + \sum_{a:\, C_a \cap S \neq \emptyset} t_a \; - \sum_a t_a \left[1 - \prod_{s \in C_a} (1 - a_s)\right] \\ &\quad = \sum_{a:\, C_a \cap S \neq \emptyset} t_a \prod_{s \in C_a} (1 - a_s) \; = \; rPD_{\mathcal{T}'_{\mathcal{X}}}(S), \end{aligned}$$

and is equal to $rPD(S)$ calculated on $\mathcal{T}'_{\mathcal{X}}$, as stated above.

It is interesting to reflect on the intuitive meaning of this transformation. The new tree is obtained by multiplying each branch by the probability that, as a result of extinctions, that branch will disappear from the tree connecting the surviving taxa. Therefore, we can think of this as a form of weighting that gives more importance to those parts in the tree that are more likely to get lost.


a)
0.1
b)
0.1
c)
E(rPD)
rPD
(expected) phylogenetic diversity
budget
0
20
50
100
150
173

For example, in Figure 4b, most of the internal branches in $\mathcal{T}'_{\mathcal{X}}$ have a length that is very close to 0. This is because these branches have many taxa below them and are therefore unlikely to be lost. Contrast this with the internal branch leading to the two Vva subspecies: this branch is relatively long, as both Vva ru and Vva va are likely to become extinct.

Thanks to this transformation, a subset of taxa that maximizes $\mathbb{E}(rPD)$ on $\mathcal{T}_{\mathcal{X}}$ subject to any given constraint also maximizes rPD on $\mathcal{T}'_{\mathcal{X}}$ subject to that same constraint, and vice versa. If the constraint is simply a limit on the number of taxa, this means that any $a_s \xrightarrow{1} 1$ NAP can be solved by solving the corresponding $0 \xrightarrow{1} 1$ NAP on the transformed tree $\mathcal{T}'_{\mathcal{X}}$, which can be done with a simple greedy algorithm (Steel, 2005; Pardi and Goldman, 2005). This result was also shown by Hartmann and Steel (2007), who independently derived the same branch length rescaling described above. The tractability of the $a_s \xrightarrow{1} 1$ NAP with a greedy algorithm also derives from its being a subproblem of the greedy-tractable $1 - q_s \xrightarrow{c} 1 - \kappa q_s$ NAP (Hartmann and Steel, 2006).

More generally, if the constraint is of the cost-budget type, the above means that any $a_s \xrightarrow{c_s} 1$ NAP can be reduced to the corresponding $0 \xrightarrow{c_s} 1$ NAP on the transformed tree $\mathcal{T}'_{\mathcal{X}}$, which can be solved with our base algorithm. Being able to solve the rather general $a_s \xrightarrow{c_s} 1$ NAP is a novel result, and we suspect that the applicability of dynamic programming techniques to the NAP is even wider.

Notice also that, even more generally, any $a_s \xrightarrow{c_s} b_s$ NAP (problem 2) can be reduced to a NAP with the form $0 \xrightarrow{c_s} b'_s$. Constructing $\mathcal{T}'_{\mathcal{X}}$ as before, a similar proof to that above shows that solving any $a_s \xrightarrow{c_s} b_s$ NAP on $\mathcal{T}_{\mathcal{X}}$ is equivalent to solving the corresponding $0 \xrightarrow{c_s} (b_s - a_s)/(1 - a_s)$ NAP on $\mathcal{T}'_{\mathcal{X}}$. Even though we cannot use this reduction to solve the general $a_s \xrightarrow{c_s} b_s$ NAP (as realistic algorithms to solve the $0 \xrightarrow{c_s} b_s$ NAP are not currently known), this shows that all the difficulty of the general NAP is somehow already present in the $0 \xrightarrow{c_s} b_s$ NAP.

It is important to realize that our dynamic programming algorithm cannot be applied or adapted to the general $a_s \xrightarrow{c_s} b_s$ NAP (or the $0 \xrightarrow{c_s} b_s$ NAP), as these problems do not have optimal substructure (it is not difficult to construct an example where the solution for a clade $\mathcal{T}$ is not obtainable as the union of solutions for $\mathcal{T}$'s subclades). One exception to this is the $0 \xrightarrow{c} b$ NAP, which does have optimal sustructure and therefore can be solved with a simple adaptation of our base dynamic programming algorithm. As the $0 \xrightarrow{c} b$ NAP can hardly have any practical importance, we do not describe this adaptation here, but this observation allows us to show that, interestingly, the class of (known) greedy-tractable subproblems of the NAP (Hartmann and Steel, 2006) is contained in the one tractable with dynamic programming algorithms: the $0 \xrightarrow{c_s} 1$ NAP applied to an ultrametric tree is a subproblem of the $0 \xrightarrow{c_s} 1$ NAP *tout court*, and any $1 - q_s \xrightarrow{c} 1 - \kappa q_s$ NAP can be reduced (following our observation in the preceding paragraph) to a corresponding $0 \xrightarrow{c} 1 - \kappa$ NAP, also solvable (as just mentioned) with dynamic programming.

Finally, note that problem (2) assumes a rooted definition of *PD*. Although we are not aware of work on the NAP using *uPD* instead of *rPD* (which we call *uPD*-NAP), it is natural to ask whether the techniques presented here (and in other papers such as Hartmann and Steel, 2006) can also be applied to the *uPD*-NAP. Using the ideas in this and the previous section, it is possible to solve any $a_s \xrightarrow{c_s} 1$ *uPD*-NAP. Its solution is either empty or equal to $\{s\} \cup R$, where R is a solution to the $a_s \xrightarrow{c_s} 1$ NAP on $\mathcal{T}^s_{\mathcal{X}}$ with budget $B - c_s$ (which we can solve as we just showed). Therefore, the problem can be solved by rooting the tree in each taxon in turn; for each rooting, transform $\mathcal{T}^s_{\mathcal{X}}$ in the way described before (this transformation is dependent on the position of the root, so we will get a different tree each time), and solve the resulting $0 \xrightarrow{c_s} 1$ NAP with budget $B - c_s$. Denoting by R_i the solutions obtained when rooting in each taxon s_i, then a solution to the $a_s \xrightarrow{c_s} 1$ *uPD*-NAP is among $\{s_1\} \cup R_1, \{s_2\} \cup R_2, \ldots, \{s_n\} \cup R_n$ and can be found by simply comparing their $\mathbb{E}(uPD)$. Note that the time complexity of this algorithm will now be $O(n^2B^2)$, as the $0 \xrightarrow{c_s} 1$ NAP will have to be solved on n different trees.

FIGURE 4. Application of our algorithms to the conservation of lemurs in Madagascar. Example for illustrative purposes only (the data are partly concocted). (a) An instance of problem (1) (the $0 \xrightarrow{c_s} 1$ NAP). A tentative phylogenetic tree of 52 lemurs (species and subspecies) was drawn on the basis of some recent publications (e.g., Yoder, 1997; Yoder et al., 2000; Pastorini et al., 2001, 2002; Roos et al., 2004; Andriaholinirina et al., 2006). The correspondence between taxa and abbreviations is reported in the Appendix. Taxon conservation costs (in the boxes next to the leaves) were estimated from information available from the IUCN Red List (IUCN, 2006) and can be considered as expressed in terms of the underlying resource of limited availability; e.g., as millions of euros. The solution of this instance for a budget $B = 20$ consists of taking the taxa in the set {Hgr gr, Ala, Iin, Lmi, Pfu el, Cme, Atr, Mmu, Dma}. Highlighted is the tree that spans these taxa. We note that this solution cannot be obtained from the one for $B = 19$ ({Hgr gr, Ala, Iin, Lmi, Pfu el, Cme, Cma, Mmu, Dma}) through a greedy step (i.e., a simple addition) but needs exchange of one taxon (Cma) for another (Atr). (b) Phylogenetic tree obtained from the one in (a) by applying the transformation described in the text for the $a_s \xrightarrow{c_s} 1$ NAP. For each taxon s, its cost c_s is indicated by the adjacent box and its probability of survival a_s by the white area in the adjacent circle. The probabilities of survival were derived from the IUCN Red List classifications (IUCN, 2006): taxa classified in risk categories CR, EN, VU, NT, LC were given probabilities 5, 25, 50, 75, and 95%, respectively. Highlighted is the tree that spans the taxa in the solution for the underlying $a_s \xrightarrow{c_s} 1$ NAP with budget $B = 20$. Again, this solution cannot be obtained through a greedy step from the one for $B = 19$ but needs exchange of one taxon (Mra) for other two taxa (Aoc and Mmy). (c) Plot showing the (expected) phylogenetic diversity of the optimal solution for the two NAP instances above as a function of the budget B. The lower and upper graph correspond to the $0 \xrightarrow{c_s} 1$ NAP and the $a_s \xrightarrow{c_s} 1$ NAP, respectively. The vertical dashed line corresponds to the budget (20) for the two solutions above, which achieve a rPD and $\mathbb{E}(rPD)$ equal to 58 and 87% of the total tree length (5.44), respectively.

DISCUSSION

Firstly, we have presented a new formalization of the problem of selecting taxa to maximize the retained phylogenetic diversity. The basic advantage, compared to past formalizations (Steel, 2005; Pardi and Goldman, 2005), is the possibility of taking into account the different resources required by the different taxa, which we assume to be quantifiable into taxon-specific costs.

Secondly, we have given algorithms that solve this problem for two definitions of *PD*. These algorithms run in $O(nB^2)$ time and use $O(nB)$ memory, where n is the number of taxa to be chosen from and B is the budget expressed as the number of units of a relevant resource (e.g., currency). The fact that the computational complexity depends on B may seem problematic: for example, if the underlying resource is money, the budget could be a very large number (of the order of the millions, if measured in common currencies) and $O(nB^2)$-time algorithms would be unusable in practice. To avoid this, B and all costs should be preprocessed and expressed as multiples of a large unit (such as their greatest common divisor, efficiently found with a generalization of Euclid's algorithm; Cormen et al., 2001). For example, in the case of conservation projects with budgets of the order of millions of euros, we may express everything in multiples of €10,000. This is probably precise enough to satisfy the accountants, makes B of the order of hundreds and permits the algorithms to run quickly. In order to check their efficiency, we implemented our algorithms (C++ code available via http://www.ebi.ac.uk/goldman/rats) and found that for values of B of the order of the thousands, our program still only takes few seconds per clade (1.7-GHz Pentium processor with 512 MB of RAM), and up to few minutes per clade for $B = 100{,}000$. Note that these are the times needed to process one clade (i.e., fill its row in the solution table) and that they do not depend on the size of clades (because each nontrivial clade will require the same $O(B^2)$ operations to be processed). The total running time is obtained by multiplying the clade-processing time by the number of clades ($2n - 1$ in the rooted case) and therefore depends on the number of taxa n but is independent of the tree shape.

Thirdly, we have shown that many cases of another formalization of taxon selection, the Noah's Ark Problem (NAP), can be transformed into instances of the problems solvable with our algorithms. We are currently working on the application of dynamic programming techniques to a form of the NAP that allows for more general relationships between conservation expenditure and probability of survival.

Although we realize that optimization problems such as the one formulated here (or even the NAP) are mainly of theoretical interest currently, it is still meaningful to ask how realistic and complete they are in including factors relevant to the selection of taxa. Many factors other than *PD* can be incorporated into problem (1), and the NAP, by adding to the length of each terminal branch a term quantifying the taxon's importance in relation to those factors (Steel, 2005; Pardi and Goldman, 2005)—this term coincides with what Weitzman (1998) calls the species' *utility*. Whereas for comparative genomics this approach seems capable of dealing with most selection criteria (Pardi and Goldman, 2005), for biodiversity conservation one of the most important factors, the actual probability of extinction, is much better dealt with by the NAP.

An important question that may be asked regarding the realism of our problem (and the NAP) is whether *PD* is a good guiding criterion for taxon selection. For comparative genomics, answering this question will involve investigating the relationship between $uPD(S)$ and the statistical power (of tests on evolutionary processes) that results from comparing the sequences in S. Along the lines already considered by Eddy (2005) and McAuliffe et al. (2005), it will be interesting to tackle questions such as: Under which circumstances is *PD* maximization not the best way to ensure high statistical power? Do these circumstances ever arise in practice? Is it possible to define an alternative measure of sequencing worth that more closely reflects power? Are the answers to these questions dependent on which statistical test is going to be carried out?

As for conservation biology, we note that *PD* is not the only measure of conservation worth that has been proposed (reviewed by Crozier, 1997). A widespread characteristic of proposed measures is that, as May recommended in his seminal note (May, 1990), they give central importance to taxonomic (phylogenetic) relationships. Because they try to formalize the intuitive notion of diversity on a phylogenetic tree, many of them are mathematically related. *Genetic diversity* (*GD*) (Crozier, 1992, 1997) is defined as the probability that the set of taxa preserves more than one allele per site. A branch a in the tree is labeled with p_a, the probability that an allele changes in the transition from one end of branch a to the other. Then $GD(S) = 1 - \prod_a (1 - p_a)$, where the product is over all branches a that are preserved (i.e., all branches in the smallest unrooted tree connecting all taxa in S). *GD* turns out to be strictly related to *uPD*: if we assume that p_a and the branch length t_a are simply related through $p_a = 1 - e^{-\kappa t_a}$, which is typical of sites where the number of alleles is so high that it is practically impossible to change back to a previously held state, then $GD(S) = 1 - \prod_a e^{-\kappa t_a} = 1 - e^{-\kappa \sum_a t_a} = 1 - e^{-\kappa \cdot uPD(S)}$. Therefore, as *GD* grows monotonically with *uPD*, maximizing *GD* is equivalent to maximizing *uPD*. This result was suggested without proof by Crozier (1997).

Even *species richness* Gaston and Spicer, 1998 (the number of different species), which is probably still the most common measure of biodiversity for conservation, is mathematically related to *PD*: it is simply equivalent to *rPD* in an ultrametric star tree. Therefore, all the results presented here can be extended to the maximization of (expected) species richness (but the optimization problems are greatly simplified and better algorithms exist). The number of taxa conserved can also be used as a secondary criterion to discriminate among equally good choices of taxa (i.e., those with equal *PD* and total cost)

and this (or even other criteria) can be easily incorporated in our algorithms by suitably setting $\iota_{\mathcal{T}}(b)$ when multiple choices for this value produce the same rPD.

Furthermore, measures of diversity mathematically equivalent to PD are likely to arise in many other fields—e.g., ecology (Petchey and Gaston, 2002a, 2002b) and social sciences (Nehring and Puppe, 2002, 2003)—whenever a tree is a good way to represent relationships (often nonevolutionary) among the objects of study. For example, a tree ("dendrogram") is often the output of data clustering techniques (Everitt et al., 2001; Eisen et al., 1998). Our results may have even wider applicability than we have suggested here.

An important shortcoming of the NAP is that species are assumed to survive or become extinct independently from one another, whereas in reality strong interdependencies may exist (for example, between predators and prey; van der Heide et al., 2005; Witting and Loeschcke, 1995; Witting et al., 2000). These dependencies can in principle be formalized, but the resulting optimization problem is likely to be a very difficult one. In particular, we doubt that dynamic programming approches such as ours will be effective: in order for a problem to have optimal substructure, a degree of independence between its parts is needed. Heuristic approaches such as hill-climbing or evolutionary algorithms may be preferable.

Note that the assumption of independence may be justified for exsitu conservation—removal and protection of a small number of individuals from their environment does not affect the survival of other taxa—but clearly not for in situ conservation (van der Heide et al., 2005). In particular, when protection is applied to a geographical area rather than to a species or population, survival probabilities are raised simultaneously for the entire group of taxa that lives in that area. Interestingly, this leads quite naturally to a generalization of the NAP in which selection is applied to a number of potential nature reserves, each defining a set of changes in survival probablities. Leaving a more precise formulation to the Appendix, we may call this the Nature Reserve Problem (NRP). When each reserve only contributes towards the survival of a single taxon, the NRP reduces to the NAP.

The problem of selecting reserves with the aim of preserving biodiversity is a well-established research topic in biodiversity conservation (e.g., Diamond, 1975; Higgs and Usher, 1980; Pressey et al., 1993; Dobson et al., 1997; Ando et al., 1998; Howard et al., 1998; Margules and Pressey, 2000; Rodrigues et al., 2004), and recently there has been a lot of interest in defining and solving (often heuristically) optimization problems for reserve selection (e.g., Margules et al., 1988; Cocks and Baird, 1989; Underhill, 1994; Camm et al., 1996; Church et al., 1996; Csuti et al., 1997; Polasky et al., 2000; Cabeza and Moilanen, 2001; Rodrigues and Gaston, 2002; Önal and Briers, 2003). Many (if not most) of the past formalizations would simply become special cases of the NRP we propose: for example, the PD-maximization problem of Rodrigues and Gaston (2002) could be seen as the $0 \xrightarrow{1} 0/1$ NRP, and the expected species richness maximization problem of Polasky et al. (2000) as the $0 \xrightarrow{1} b_{rs}$ NRP applied to an ultrametric star tree (see Appendix for the definition of the notation used here). The NRP would thus unify the (arguably most successful) approaches from economics and biology to the problem of biodiversity conservation, thus becoming a fertile meeting ground for the exchange of new ideas between these two disciplines.

ACKNOWLEDGMENTS

We thank Klaas Hartmann, David MacKay, and two anonymous reviewers for their many helpful comments. F.P. is a member of St Catharine's College, University of Cambridge.

REFERENCES

Ando, A., J. Camm, S. Polasky, and A. Solow. 1998. Species distributions, land values, and efficient conservation. Science 279:2126–2128.

Andriaholinirina, N., J. Fausser, C. Roos, D. Zinner, U. Thalmann, C. Rabarivola, I. Ravoarimanana, J. Ganzhorn, B. Meier, R. Hilgartner, L. Walter, A. Zaramody, C. Langer, T. Hahn, E. Zimmermann, U. Radespiel, M. Craul, J. Tomiuk, I. Tattersall, and Y. Rumpler. 2006. Molecular phylogeny and taxonomic revision of the sportive lemurs (Lepilemur, Primates). BMC Evol. Biol. 6:17.

Barker, G. 2002. Phylogenetic diversity: A quantitative framework for measurement of priority and achievement in biodiversity conservation. Biol. Linn. Soci. 76:165–194.

Cabeza, M., and A. Moilanen. 2001. Design of reserve networks and the persistence of biodiversity. Trends Ecol. Evol. 16:242–248.

Camm, J., S. Polasky, A. Solow, and B. Csuti. 1996. A note on optimal algorithms for reserve site selection. Biol. Conserv. 78:353–355.

Church, R., D. Stoms, and F. Davis. 1996. Reserve selection as a maximal covering location problem. Biol. Conserv. 76:105–112.

Cocks, K., and I. Baird. 1989. Using mathematical programming to address the multiple reserve selection problem: An example from the Eyre Peninsula, South Australia. Biol. Conserv. 49:113–130.

Cormen, T., C. Leiserson, R. Rivest, and C. Stein. 2001. Introduction to algorithms, 2nd edition. MIT Press, Cambridge, Massachusetts.

Crozier, R. 1992. Genetic diversity and the agony of choice. Biol. Conserv. 61:11–15.

Crozier, R. 1997. Preserving the information content of species: Genetic diversity, phylogeny, and conservation worth. Annu. Rev. Ecol. Syst. 28:243–268.

Crozier, R., P. Agapow, and L. Dunnett. 2006. Conceptual issues in phylogeny and conservation: A reply to Faith and Baker. Evol. Bioinformatics Online 2:197–199.

Csuti, B., S. Polasky, P. Williams, R. Pressey, J. Camm, M. Kershaw, A. Kiester, B. Downs, R. Hamilton, M. Huso, and K. Sahr. 1997. A comparison of reserve selection algorithms using data on terrestrial vertebrates in Oregon. Biol. Conserv. 80:83–97.

Diamond, J. 1975. The island dilemma: lessons of modern biogeographic studies for the design of natural reserves. Biol. Conserv. 7:129–146.

Dobson, A., J. Rodriguez, W. Roberts, and D. Wilcove. 1997. Geographic distribution of endangered species in the United States. Science 275:550–553.

Eddy, S. 2005. A model of the statistical power of comparative genome sequence analysis. PLoS Biol. 3:e10.

Eisen, M., P. Spellman, P. Brown, and D. Botstein. 1998. Cluster analysis and display of genome-wide expression patterns. Proc. Nat. Acad. Sci. USA 95:14863–14868.

Everitt, B., S. Landau, and M. Leese. 2001. Cluster analysis, 4th edition. Arnold, London.

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

Faith, D., and A. Baker. 2006. Phylogenetic diversity (PD) and biodiversity conservation: Some bioinformatics challenges. Evol. Bioinformatics Online 2:70–77.

Forest, F., R. Grenyer, M. Rouget, T. Davies, R. Cowling, D. Faith, A. Balmford, J. Manning, Ş. Procheş, M. van der Bank, G. Reeves, T. Hedderson, and V. Savolainen. 2007. Preserving the evolutionary potential of floras in biodiversity hotspots. Nature 445:757–760.

Garey, M. and D. Johnson. 1979. Computers and intractability: A guide to the theory of NP-completeness. W.H. Freeman and Co., New York.

Gaston, K. and J. Spicer. 1998. Biodiversity: An introduction. Blackwell Science, Oxford, UK.

Hartmann, K., and M. Steel. 2006. Maximizing phylogenetic diversity in biodiversity conservation: Greedy solutions to the Noah's Ark Problem. Syst. Biol. 55:644–651.

Hartmann, K., and M. Steel. 2007. Phylogenetic diversity: From combinatorics to ecology. in Reconstructing evolution: New mathematical and computational advances (O. Gascuel and M. Steel, eds.). Oxford University Press.

Higgs, A., and M. Usher. 1980. Should nature reserves be large or small? Nature 285:568–569.

Howard, P., P. Viskanic, T. Davenport, F. Kigenyi, M. Baltzer, C. Dickinson, J. Lwanga, R. Matthews, and A. Balmford. 1998. Complementarity and the use of indicator groups for reserve selection in Uganda. Nature 394:472–475.

IUCN. 2006. 2006 IUCN Red List of Threatened Species. http://www.iucnredlist.org (accessed 30 December 2006).

Khuller, S., A. Moss, and J. Naor. 1999. The budgeted maximum coverage problem. Inform. Process. Lett. 70:39–45.

Mace, G., J. Gittleman, and A. Purvis. 2003. Preserving the Tree of Life. Science 300:1707.

Margules, C., A. Nicholls, and R. Pressey. 1988. Selecting networks of reserves to maximize biological diversity. Biol. Conserv. 43:63–76.

Margules, C., and R. Pressey. 2000. Systematic conservation planning. Nature 405:243–253.

Margulies, E., J. Vinson, W. Miller, D. Ja–e, K. Lindblad-Toh, J. Chang, E. Green, E. Lander, J. Mullikin, and M. Clamp. 2005. An initial strategy for the systematic identification of functional elements in the human genome by low-redundancy comparative sequencing. Proc. Nat. Acad. Sci. USA 102:4795–4800.

May, R. 1990. Taxonomy as destiny. Nature 347:129–130.

McAuliffe, J., M. Jordan, and L. Pachter. 2005. Subtree power analysis and species selection for comparative genomics. Proc. Nat. Acad. Sci. USA 102:7900–7905.

Minh, B., S. Klaere, and A. von Haeseler. 2006. Phylogenetic diversity within seconds. Syst. Biol. 55:769–773.

Moritz, C., and D. Faith. 1998. Comparative phylogeography and the identification of genetically divergent areas for conservation. Mol. Ecol. 7:419–429.

Nee, S., and R. May. 1997. Extinction and the loss of evolutionary history. Science 278:692–694.

Nehring, K., and C. Puppe. 2002. A theory of diversity. Econometrica 70:1155–1198.

Nehring, K., and C. Puppe. 2003. Diversity and dissimilarity in lines and hierarchies. Math. Social Sci. 45:167–183.

Önal, H., and R. Briers. 2003. Selection of a minimum-boundary reserve network using integer programming. Proc. R. Soc. B Biol. Sci. 270:1487–1491.

Pardi, F., and N. Goldman. 2005. Species choice for comparative genomics: No need for cooperation. PLoS Genetics 1:e71.

Pastorini, J., M. Forstner, and R. Martin. 2002. Phylogenetic relationships among Lemuridae (Primates): Evidence from mtDNA. J. Hum. Evol. 43:463–478.

Pastorini, J., R. Martin, P. Ehresmann, E. Zimmermann, and M. Forstner. 2001. Molecular phylogeny of the lemur family Cheirogaleidae (Primates) based on mitochondrial DNA sequences. Mol. Phylogenet. Evol. 19:45–56.

Petchey, O., and K. Gaston. 2002a. Extinction and the loss of functional diversity. Proc. R. Soc. B Biol. Sci. 269:1721–1727.

Petchey, O., and K. Gaston. 2002b. Functional diversity (FD), species richness and community composition. Ecol. Lett. 5:402–411.

Polasky, S., J. Camm, A. Solow, B. Csuti, D. White, and R. Ding. 2000. Choosing reserve networks with incomplete species information. Biol. Conserva. 94:1–10.

Pressey, R., C. Humphries, C. Margules, R. Vane-Wright, and P. Williams. 1993. Beyond opportunism: Key principles for systematic reserve selection. Trends Ecol. Evol. 8:124–128.

Reist-Marti, S., H. Simianer, J. Gibson, O. Hanotte, and J. Rege. 2003. Weitzman's approach and conservation of breed diversity: An application to African cattle breeds. Conserv. Biol. 17:1299–1311.

Rodrigues, A., S. Andelman, M. Bakarr, L. Boitani, T. Brooks, R. Cowling, L. Fishpool, G. da Fonseca, K. Gaston, M. Hoffmann, J. S. Long, P. A. Marquet, J. D. Pilgrim, R. L. Pressey, J. Schipper, W. Sechrest, S. N. Stuart, L. G. Underhill, R. W. Waller, M. E. J. Watts, and X. Yan. 2004. Effectiveness of the global protected area network in representing species diversity. Nature 428:640–643.

Rodrigues, A., and K. Gaston. 2002. Maximizing phylogenetic diversity in the selection of networks of conservation areas. Biol. Conserv. 105:103–111.

Roos, C., J. Schmitz, and H. Zischler. 2004. Primate jumping genes elucidate strepsirrhine phylogeny. Proc. Nat. Acad. Sci. USA 101:10650–10654.

Simianer, H., S. Marti, J. Gibson, O. Hanotte, and J. Rege. 2003. An approach to the optimal allocation of conservation funds to minimize loss of genetic diversity between livestock breeds. Ecol. Econ. 45:377–392.

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

Thomas, J., J. Touchman, R. Blakesley, G. Bouffard, S. Beckstrom-Sternberg, E. Margulies, M. Blanchette, A. Siepel, P. Thomas, J. McDowell, et al. 2003. Comparative analyses of multi-species sequences from targeted genomic regions. Nature 424:788–793.

Underhill, L. 1994. Optimal and suboptimal reserve selection algorithms. Biol. Conserv. 70:85–87.

van der Heide, C., J. van den Bergh, and E. van Ierland. 2005. Extending Weitzman's economic ranking of biodiversity protection: Combining ecological and genetic considerations. Ecol. Econ. 55:218–223.

Vane-Wright, R., C. Humphries, and P. Williams. 1991. What to protect?—Systematics and the agony of choice. Biol. Conserv. 55:235–254.

Weitzman, M. 1992. On diversity. Q. J. Econ. 107:363–405.

Weitzman, M. 1993. What to preserve? An application of diversity theory to crane conservation. Q. J. Econ. 108:157–183.

Weitzman, M. 1998. The Noah's Ark Problem. Econometrica 66:1279–1298.

Witting, L., and V. Loeschcke. 1993. Biodiversity conservation: Reserve optimization or loss minimization? Trends Ecol. Evol. 8:417.

Witting, L., and V. Loeschcke. 1995. The optimization of biodiversity conservation. Biol. Conserv. 71:205–207.

Witting, L., J. Tomiuk, and V. Loeschcke. 2000. Modelling the optimal conservation of interacting species. Ecol. Model. 125:123–143.

Yoder, A. 1997. Back to the future: A synthesis of strepsirrhine systematics. Evol. Anthropol. Issues News Rev. 6:11–22.

Yoder, A., R. Rasoloarison, S. Goodman, J. Irwin, S. Atsalis, M. Ravosa, and J. Ganzhorn. 2000. Remarkable species diversity in Malagasy mouse lemurs (primates, Microcebus). Proc. Nat. Acad. Sci. USA 97:11325–11330.

First submitted 21 September 2006; reviews returned 16 November 2006; final acceptance 29 December 2006
Associate Editor: Mike Steel

APPENDIX

The Algorithm for the Unrooted Case

Here, we define a *clade* as any rooted subtree $\mathcal{T}$ of $\mathcal{T}_{\mathcal{X}}$ consisting of (a) a branch a, with one of its ends being the root of $\mathcal{T}$, and (b)

everything else in $\mathcal{T}_\mathcal{X}$ that lies on the other (i.e., nonroot) side of a. This definition includes all the clades in the previous sense, but whereas before each branch identified one single clade, now for each branch there are two clades, each rooted by one of its ends. For example, the branch of length 3.0 in Figure 3 is at the "top" of both clades (A,B) and (C,(D,E)). Assuming that $\mathcal{T}_\mathcal{X}$ is originally unrooted (the position of any root has no relevance for *uPD*), there are exactly $2(2n-3)$ clades in $\mathcal{T}_\mathcal{X}$. In particular, note that the $\mathcal{T}_\mathcal{X}^s$, the versions of $\mathcal{T}_\mathcal{X}$ rooted in each taxon s, are now clades.

Although the collection of clades is now nonhierarchical, the fundamental properties of a single clade have not changed: a clade is still a rooted tree with one branch departing from its root and with its leaves being a subset of the leaves of $\mathcal{T}_\mathcal{X}$. Therefore, the rooted subproblems defined before for all clades $\mathcal{T}$ and sub-budgets $b \leq B$ remain well defined. We are particularly interested in solving the subproblems for the taxon-rooted clades $\mathcal{T}_\mathcal{X}^s$ and sub-budgets $B - c_s$ because, as previously observed, some of their solutions R_s provide us with solutions $\{s\} \cup R_s$ to the unrooted problem.

All the rooted subproblems will again be implicitly solved by incrementally deriving $\lambda_\mathcal{T}(b)$ and $\iota_\mathcal{T}(b)$ for all $\mathcal{T}$ and b. As we showed before, the calculation of these values is either straightforward or directly obtainable from the corresponding values for $\mathcal{T}$'s subclades. It is therefore necessary to tackle the clades in an order that guarantees that subclades are met before their "superclades." Whereas before this was trivially satisfied by a bottom-up traversal of all clades, now a slightly more complex approach is needed. First, a node in $\mathcal{T}_\mathcal{X}$ is arbitrarily chosen as *top node* of the tree (different choices of this node, which can be a leaf, lead to different orderings of the clades, but all produce the same final results). This determines a way to picture the tree (imagine it redrawn with the top node uppermost and all branches descending from there) and we can then classify clades into *downward* clades—those consisting of a branch and everything below it—and *upward* clades—the remaining ones. For example, in Figure 3 the top node is set to the position highlighted in pink; as a result, (A,B) is a downward clade and (C,(D,E)) an upward clade.

The λ and ι values are calculated for downward clades first, going in the usual bottom-up order. In Figure 3, where again we imagine that all the results are stored in a solutions table, this corresponds to filling the bottom half of the table, starting from the bottom row and going upwards. The results for the upward clades are then derived in a top-down fashion: more precisely, we can imagine that all the branches in $\mathcal{T}_\mathcal{X}$ are visited in a top-down order and each time a branch is visited, the λ and ι values for the corresponding upward clade are computed. In Figure 3, upward clades are visited in the following order: 1: (C,(D,E)); 2: (B,(C,(D,E))); 3: (A,(C,(D,E))); 4: ((A,B),(D,E)); 5: ((A,B),C); 6: (((A,B),C),E); 7: (((A,B),C),D), which corresponds to filling the top half of the solutions table from bottom to top. This ordering of the clades guarantees that whenever we derive solutions for a clade $\mathcal{T}$ containing subclades $\mathcal{L}$ and $\mathcal{R}$, the solutions for $\mathcal{L}$ and $\mathcal{R}$ have already been calculated.

Once all the λ and ι values have been calculated, we turn our attention to the values $\lambda_{\mathcal{T}_\mathcal{X}^s}(B - c_s)$ for all taxa s. By definition, these are equal to $rPD(R_s)$, where R_s is a solution to the rooted problem with budget $B - c_s$ applied to $\mathcal{T}_\mathcal{X}^s$, and therefore also equal to the *uPD* of the candidate solutions $\{s\} \cup R_s$ to the unrooted problem. Clearly, the taxa s that maximize $\lambda_{\mathcal{T}_\mathcal{X}^s}(B - c_s)$ are those contained in an optimal solution to the unrooted problem (in Figure 3, all of them except B). If we pick any one of these taxa s and reconstruct R_s by using the $\iota_\mathcal{T}$ values starting from $\iota_{\mathcal{T}_\mathcal{X}^s}(B - c_s)$—as described for the rooted case or, equivalently, with a call to RECONSTRUCT($\mathcal{T}_\mathcal{X}^s$, $B - c_s$)—we therefore also obtain an optimal solution $\{s\} \cup R_s$ to the unrooted problem. For example, if in Figure 3 we pick taxon C, following the $\iota_\mathcal{T}$ values (indicated in the figure by dotted arrows) leads to rooted solution {A,D,E} and therefore {A,C,D,E} is a solution to the unrooted problem.

Note that whereas in this example {A,C,D,E} coincides with the set of taxa that maximize $\lambda_{\mathcal{T}_\mathcal{X}^s}(B - c_s)$, in general the latter will not necessarily coincide with an optimal solution but rather with the *union* of all optimal solutions to the unrooted problem.

This concludes the description of our algorithm for maximizing *uPD* subject to cost constraints. Again, a more concise description is given by the pseudocode in algorithms 1–5: a solution to the unrooted problem is simply obtained by a call to DOUBLE TRAVERSAL($\mathcal{T}_\mathcal{X}$). The cheapest optimal solution can be obtained in a similar way to that described above: just try to reduce the budget below $B - c_s$ for all taxa s that maximize $\lambda_{\mathcal{T}_\mathcal{X}^s}(B - c_s)$; one of the taxa that lead to the largest reduction should be used as starting point for the reconstruction of the minimal-cost *uPD*-optimal set of taxa.

Pseudocode

Problem (1) with the rooted definition for *PD* is solved by a call to BOTTOM-UP($\mathcal{T}_\mathcal{X}$), which fills up the solutions table, followed by a call to RECONSTRUCT($\mathcal{T}_\mathcal{X}$, B), which returns an optimal solution (not necessarily of minimum cost, but this can easily be implemented).

Problem (1) with the unrooted definition for *PD* is solved by a call to DOUBLE TRAVERSAL($\mathcal{T}_\mathcal{X}$) (where $rev(\mathcal{T})$ denotes the clade rooted in the same branch as $\mathcal{T}$ but oriented towards the opposite side).

A simple improvement to these algorithms can be obtained by noting that the rows of the solutions table do not always need to be filled up until their last column. If a clade $\mathcal{T}$ is such that the total sum of the costs for its taxa, which we may denote by $C_\mathcal{T}$, is smaller than B, then its $\lambda_\mathcal{T}(b)$ and $\iota_\mathcal{T}(b)$ values need only be calculated for $b \leq C_\mathcal{T}$: the solutions for $b > C_\mathcal{T}$ are necessarily the same as the one for $b = C_\mathcal{T}$ and simply consist of taking all taxa in $\mathcal{T}$. Procedures CALCULATE($\mathcal{T}$) and RECONSTRUCT($\mathcal{T}$, b) can be simply modified accordingly. This results in some saving of running time, although the time complexity remains $O(nB^2)$.

Algorithm 1 BOTTOM-UP($\mathcal{T}$)

if $\mathcal{T}$ contains subclades $\mathcal{L}$ and $\mathcal{R}$ **then**
  BOTTOM-UP($\mathcal{L}$)
  BOTTOM-UP($\mathcal{R}$)
end if
CALCULATE($\mathcal{T}$)

Algorithm 2 CALCULATE($\mathcal{T}$)

if $\mathcal{T}$ only consists of a terminal branch a ending in taxon s **then**
  $\check{c}_\mathcal{T} = c_s$
  for $b = 0, \ldots, \check{c}_\mathcal{T} - 1$ **do** $\lambda_\mathcal{T}(b) = 0$
  for $b = \check{c}_\mathcal{T}, \ldots, B$ **do** $\lambda_\mathcal{T}(b) = t_a$
end if
if $\mathcal{T}$ consists of an internal branch a and subclades $\mathcal{L}$ and $\mathcal{R}$ **then**
  $\check{c}_\mathcal{T} = \min\{\check{c}_\mathcal{L}, \check{c}_\mathcal{R}\}$
  for $b = 0, \ldots, \check{c}_\mathcal{T} - 1$ **do** $\lambda_\mathcal{T}(b) = 0$, $\iota_\mathcal{T}(b) = 0$
  for $b = \check{c}_\mathcal{T}, \ldots, B$ **do**
    $\iota_\mathcal{T}(b) = \arg\max_{i \in \{0,\ldots,b\}} \{\lambda_\mathcal{L}(i) + \lambda_\mathcal{R}(b - i)\}$
    $\lambda_\mathcal{T}(b) = t_a + \lambda_\mathcal{L}(\iota_\mathcal{T}(b)) + \lambda_\mathcal{R}(b - \iota_\mathcal{T}(b))$
    if $\lambda_\mathcal{L}(\iota_\mathcal{T}(b)) + \lambda_\mathcal{R}(b - \iota_\mathcal{T}(b)) = 0$ **then**
      if $\check{c}_\mathcal{L} \leq b$ **then** $\iota_\mathcal{T}(b) = \check{c}_\mathcal{L}$ **else** $\iota_\mathcal{T}(b) = b - \check{c}_\mathcal{R}$
    end if
  end for
end if

Algorithm 3 RECONSTRUCT($\mathcal{T}$, b)

if $\mathcal{T}$ only consists of a terminal branch ending in taxon s **then**
  if $b \geq c_s$ **return** $\{s\}$
  if $b < c_s$ **return** $\emptyset$
end if
if $\mathcal{T}$ contains subclades $\mathcal{L}$ and $\mathcal{R}$ **then**
  return RECONSTRUCT($\mathcal{L}$, $\iota_\mathcal{T}(b)$) $\cup$ RECONSTRUCT($\mathcal{R}$, $b - \iota_\mathcal{T}(b)$)
end if

Algorithm 4 DOUBLE TRAVERSAL($\mathcal{T}$)

do root $\mathcal{T}$ in any of its leaves
BOTTOM-UP($\mathcal{T}$)
TOP-DOWN($rev(\mathcal{T})$)
let s be a leaf of $\mathcal{T}$ that maximizes $\lambda_{\mathcal{T}^s}(B - c_s)$
return $\{s\} \cup$ RECONSTRUCT($\mathcal{T}^s$, $B - c_s$)

Algorithm 5 TOP-DOWN($\mathcal{T}$)

CALCULATE($\mathcal{T}$)
if $\mathcal{T}$ is a subclade of two other clades $\mathcal{F}$ and $\mathcal{M}$ **then**
    TOP-DOWN($\mathcal{F}$)
    TOP-DOWN($\mathcal{M}$)
end if

The Nature Reserve Problem

Imagine having, in addition to the species in $\mathcal{X}$, a set $\mathcal{R}$ of potential reserve sites. A pair (r, s), with $r \in \mathcal{R}$ and $s \in \mathcal{X}$, identifies the (possibly nonexistent) population of species s in site r. Imagine that we have control over the matrix (p_{rs}) of survival probabilities of the populations (r, s) (where by "survival" we mean existence at some specified future time). In analogy to the Noah's Ark Problem, these probabilities can be chosen from two specified values: we have two matrices of probabilities (a_{rs}), (b_{rs}), and p_{rs} will be set to b_{rs} or a_{rs} depending on whether site r is conserved or not, respectively. Note that we may have $a_{rs} > b_{rs}$, when for example species s may benefit from the human exploitation of site r. For each $r \in \mathcal{R}$, we also have a measure c_r of the cost of conserving site r; i.e., of including it in a nature reserve. The objective is to construct a set of reserves $R \subseteq \mathcal{R}$, with $\sum_{r \in R} c_r \leq B$, that maximizes the expected *PD* resulting from conservation of the sites in R.

The Nature Reserve Problem, which we may write $a_{rs} \xrightarrow{c_r} b_{rs}$ NRP, is a generalization of many problems defined in the past. In addition to the ones already mentioned, we note that the Budgeted Maximum Coverage Problem (Khuller et al., 1999) coincides with the $0 \xrightarrow{c_r} 0/1$ NRP applied to a star tree—where by writing $0/1$ on the right of the arrow we mean that each survival probability b_{rs} is constrained to equal 0 or 1.

Taxa in Figure 4

Species abbreviations are Dma: *Daubentonia madagascariensis*; Mmy: *Microcebus myoxinus*; Mru: *Microcebus rufus*; Mra: *Microcebus ravelobensis*; Mmu: *Microcebus murinus*; Mco: *Mirza coquereli*; Atr: *Allocebus trichotis*; Cma: *Cheirogaleus major*; Cme: *Cheirogaleus medius*; Pfu el: *Phaner furcifer electromontis*; Pfu pl: *Phaner furcifer pallescens*; Pfu pr: *Phaner furcifer parienti*; Pfu fu: *Phaner furcifer furcifer*; Lmu: *Lepilemur mustelinus*; Ldo: *Lepilemur dorsalis*; Lse: *Lepilemur septentrionalis*; Led: *Lepilemur edwardsi*; Lmi: *Lepilemur microdon*; Lru: *Lepilemur ruficaudatus*; Lle: *Lepilemur leucopus*; Pta: *Propithecus tattersalli*; Pco: *Propithecus coquereli*; Pve ve: *Propithecus verreauxi verreauxi*; Pve de: *Propithecus verreauxi deckeni*; Pve co: *Propithecus verreauxi coronatus*; Pdi: *Propithecus diadema*; Ppe: *Propithecus perrieri*; Pca: *Propithecus candidus*; Ped: *Propithecus edwardsi*; Iin: *Indri indri*; Ala: *Avahi laniger*; Aoc: *Avahi occidentalis*; Acl: *Avahi cleesei*; Vva va: *Varecia variegata variegata*; Vva ru: *Varecia variegata rubra*; Hgr al: *Hapalemur griseus alaotrensis*; Hgr oc: *Hapalemur griseus occidentalis*; Hgr gr: *Hapalemur griseus griseus*; Hau: *Hapalemur aureus*; Hsi: *Hapalemur simus*; Lca: *Lemur catta*; Ema ma: *Eulemur macaco macaco*; Ema fl: *Eulemur macaco flavifrons*; Eco: *Eulemur coronatus*; Eru: *Eulemur rubriventer*; Emo: *Eulemur mongoz*; Efu co: *Eulemur fulvus collaris*; Efu ac: *Eulemur fulvus albocollaris*; Efu ru: *Eulemur fulvus rufus*; Efu fu: *Eulemur fulvus fulvus*; Efu af: *Eulemur fulvus albifrons*; Efu sa: *Eulemur fulvus sanfordi*. The phylogenetic tree used in Figure 4a and the species' conservation costs and survival probabilities are available from http://www.ebi.ac.uk/goldman/rats.