WILEY-BLACKWELL

Cladistics 24 (2008) 813–824

Cladistics

10.1111/j.1096-0031.2008.00203.x

What is PACT really?

J. Salvador Arias[a], Ivonne J. Garzón-Orduña[a,b], Federico López-Osorio[a], Erika Parada-Vargas[a] and Daniel Rafael Miranda-Esquivel[a]*

[a]*Laboratorio de Sistemática & Biogeografía, Escuela de Biología, Universidad Industrial de Santander, A.A. 678 Bucaramanga, Colombia;* [b]*Department of Biological Sciences, University of New Orleans, 2000 Lakeshore Drive, New Orleans, LA, 70148, USA*

Accepted 30 October 2008

Abstract

"Phylogenetic Analysis for Comparing Trees" (PACT) has been presented as a "new algorithm" for the study of biogeography and coevolution. However, an exploration of this algorithm revealed some important problems missed in the original description. First, PACT is not new, rather it is an extension of Tree Mapping under Maximum Codivergence (TM-MC). Second, as was described, PACT lacks an optimality criterion, and like secondary BPA, it does not offer a solution for handling incongruent elements. We found that PACT and TM-MC differ only in the way the final answer is presented, and in the absence of an explicit algorithm of historical reconstruction under PACT. Given the equivalence between TM-MC and PACT in their aims and assumptions, the criticism to TM-MC as "orthogenetic" is not well founded.

Brooks parsimony analysis (BPA; Brooks, 1981) is one of the most used and yet one of the most heavily criticized approaches in historical biogeography/host–parasite analyses. To overcome the problems of BPA, Wojcicki and Brooks (2004, 2005; WB henceforth)[1] presented the "Phylogenetic Analysis for Comparing Trees" (PACT), as a "new algorithm" for generating host or area cladograms from trees representing the association between parasites and hosts or taxa and areas. Here we show that PACT and Tree Mapping under Maximum Cospeciation (TM-MC; Page, 1994b) are equivalent. First we present a general definition of both algorithms and then we show empirical examples that led to criticisms of PACT.

In contrast with BPA, PACT uses lineages rather than a tree representation matrix (hence "escaping" the matrix), and shares with secondary BPA (SBPA; Brooks, 1990) the duplication of associates that have a reticulated history. However, according to WB, PACT does not manipulate the data a posteriori. Just like SBPA, PACT intends "to represent clearly all exceptions to […] [the] general pattern" (Brooks et al., 2001, p. 349) [i.e., to find "partial congruence" (WB, p. 352/766)].

Tree reconciliation (Goodman et al., 1979; Page, 1994a), and its derivation TM-MC (Page, 1994b) is based on the concept of a map between trees. A map is a function that assigns each node of a given tree (for example, the parasite tree) to a node in another associate tree (for example, the host tree). To find the map between trees (Table 1; Fig. 1), a terminal-associate cladogram (hereafter TAC) must be constructed replacing the original terminal label (the parasite or taxon) with the label of the host(s) or area(s) associated with each terminal (the associate). The set of associates of each node in the TAC is constructed adding the set of associates of its descendants with a post-order traverse of the tree. The image in the other tree (for example, the associate tree) is determined by matching the associate sets. If a given node of the TAC, say "*n*", matches a given node of the second tree, say "*m*", and no

*Corresponding author:
E-mail address: dmiranda@uis.edu.co

[1]As both papers are nearly identical they are cited as only one paper. The first page, table, or figure cited corresponds to the paper published in 2004. The second one, separated by a slash ("/") refers to the paper published in 2005.

Table 1
The TM-MC algorithm, modified from Page (1994b). The notation of template and input tree is taken from Wojcicki and Brooks (2004, 2005)

1.	Choose one cladogram, the template tree.
2.	Let the *l*-node be the parasite node that switched associates, the s-node be the parasite node that remained on an ancestral associate, and the *j*-node be the immediate ancestor of at most one *s*-node (see Fig. 1).
3.	Define the associate sets of each terminal.
4.	Define the associate sets for each internal node, as the union of each descendant node associate sets except *l*-nodes.
5.	Choose another cladogram as the input tree and perform steps 2–4 using it.
6.	For all nodes in the input tree (except any *j*-node) find the image of their associate set in the template tree. This defines the map between template and input tree.
7.	Let the matched node be a node a from the input tree that has as image the node *m* in the template tree, in which any of its descendants has as images a node different from *m*.
8.	Choose the map(s) that maximize(s) the number of matching nodes.

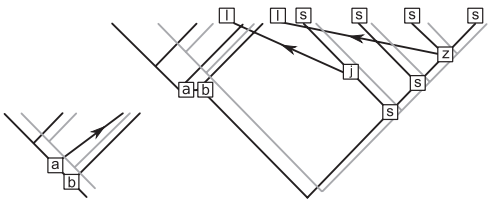

Fig. 1. Terminology of TM-MC. Node *l* represents a parasite that switched associates, node *s* represents a node that remains on the ancestral associate, node *j* represents a node that contains at most only one *s* node and other *l* and/or *j* nodes. The node marked with a z (a trichotomy) is an *s* node, not a *j* node. The insert shows a possible reconstruction for a duplication, -a- node on the main image represents a case of codivergence and node -b- is a case of duplication. In the insert, node -a- is a *j* node whereas node -b- is inferred as a codivergence. In both cases, a codivergence event and the same reconstruction (in terms of topology) is found, although given the duplication of associates, the inferred events are ambiguous. Modified from Page (1994b).

descendant of *n* matches *m*, the match is counted as a codivergence (cospeciation); otherwise, it is inferred as a duplication. Given that many reconstructions (i.e., maps) are possible, Page suggested "reconstructions that maximize the number of [codivergences] have the greatest explanatory power and are hence preferred over reconstructions with fewer [codivergences]" (Page, 1994b, p. 155). A duplication could also be interpreted as a dispersion, without changing the reconstruction or the number of codivergences (see Fig. 1).

The above-mentioned tree-mapping methods have been strongly criticized by Brooks and his collaborators (Brooks, 2003; Brooks et al., 2004; Brooks and Ferrao, 2005). Strikingly, WB never compared their own algorithm with these methods.

Equivalence of PACT and TM-MC

Equivalence of algorithm

WB did not present the mechanics of their algorithm in a comprehensive way; they use a summary table (their table 2) along with a series of examples. This makes it impossible to know how the algorithm works in cases not accounted for in their examples. Here we follow the steps in their table 2 (in Appendix 1 we attempt a complete re-formalization of the PACT algorithm).

Steps 1 and 2 are not relevant to the final answer. All association methods start with a TAC and as a TAC is a tree, it can be represented by a Venn diagram. It seems WB meant that TACs must be input using parenthetical notation; but the notation of the input is not relevant for the algorithm.

We must emphasize that in the construction of the TAC, the set of associates that belong to broad range parasites is assumed to be part of a single node (i.e., the reconstruction of TACs follows assumption 0). Though stressed in other Brooks' works (e.g., Brooks and Ferrao, 2005), this caution is included in the text but not in WB's table 2.

In step 3, a starting tree, the *template cladogram*, is chosen. According to WB, any tree can be selected *without* changing the results, but in some cases (even in the ones presented by WB) this does not apply (see below). Next, the elements of the template tree are determined using a post-order traversal.[2] There is some ambiguity about how the data are stored. Following their example 1 (WB, pp. 342–346/757–761), elements are stored as clades, that is, the set of associates retain a hierarchical structure. But in a more complex situation, such as example 2 (WB, pp. 346–348/761–763), the data are stored as a string (the list of associates with no hierarchy).

The next step in PACT is to choose another tree (WB, table 2), called the *input tree*. A set of associates is constructed in the same way it was done for the nodes in the template tree.

In the corresponding steps of TM-MC (Page, 1994a,b), all trees are "converted" to TACs using the association set of each terminal and the set of associates, a string with the list of associates, is determined using post-order traversal. (Note that WB used the term "elements of the tree" for Page's "set of associates".) In PACT as in TM-MC at the end of these steps we have a TAC with the sets of associates for each node.

Subsequently in PACT, TACs are successively parsed and merged to the "template tree". In each step, the algorithm operates in two trees: the template tree and

[2]WB's implementation requires at least two passes, because elements are read after each closed parenthesis is found. A recursive implementation is feasible.

the tree with which it is to be merged, the "input" TAC. In the original description of TM-MC, TACs are not merged; instead, a mapping between TACs is constructed. But once a map is constructed, the merging of two TACs is straightforward (see next section).

To determine the merge, WB provided several rules. If an associate set of a node in the template tree is identical to some associate set in the input tree, then the merged tree includes that node, this is the Y + Y = Y rule. However, if the contents of compared associate sets are only partly identical (i.e., some associates are absent in one of the two nodes), then a second (Y + YN = YN)[3] or a third rule (YN + YN = YNN) are invoked, respectively. Under the latter rules the merged tree includes a node containing all the elements from the two trees. To avoid trivial matching the three WB rules must be hierarchical or nearly any match could be justified with the YN + YN = YNN rule; this seems to be implied in the way each rule is presented.

In TM-MC (Page, 1994b, p. 159) matches are found as "the smallest set containing *all* elements of the [associate set]" (italics in the original). As Page deals with parasites and their associates, he does not explicitly present a parasite-without-associate situation, so it is equivalent to Y + Y = Y, and Y + YN = YN. Furthermore in both, PACT and TM-MC, when associate sets are unique to either of the trees they are not used in determining the map, but are added to the equivalent node in the final stage of fusion.

Sometimes the TACs under comparison have incongruent groups; for example, a group supported in one of the TACs could be found contradicted in the other TAC. A closer examination on each clade might show though that some nodes are combinable, if one or some more elements are ignored. This is the rationale of the removal of associates proposed under TM-MC (Page, 1994b, p. 159). The removal of an associate does not imply that the node is really eliminated; on the contrary it provides a hypothesis that the node (or terminal) associated with the element removed has a different history in both clades under comparison. At first look, it seems like if WB took another solution: they label the problematic (incongruent) element as a new one (WB, p. 344/759). However, this is exactly the same procedure used by TM-MC: as it is considered a new element, the incongruent element is eliminated from the associate set (Fig. 2). The appearance of a newly created element implies a different history between the compared clades for that association.

In the next step of the PACT algorithm a fourth rule is introduced, "Y(Y– = Y(Y–", which is considered a novel rule by WB: "[a]ll current methods, including

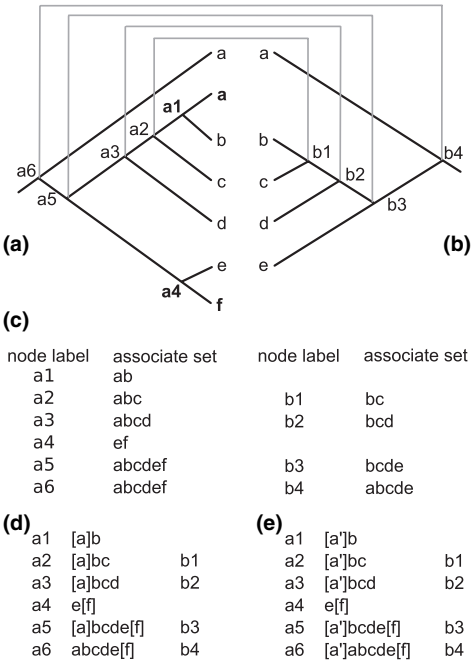

Fig. 2. Comparison between TM-MC and PACT when dealing with novel elements, and associate switched terminals. (a,b) The trees to be combined, gray lines represent the map between internal nodes and nodes in bold represent new elements; (c) host set at each node of both trees, second -a- is redundant in node a6, therefore is not included in the associate list; (d) the map using TM-MC algorithm, associate [a] is not counted, it is inferred as an associate-switched terminal, and associate [f] is not counted as it is present only in one tree; (e) the map using the PACT algorithm, the node [a] inferred as an associate-switched terminal is treated as a new element [a'] and like associate [f] is not counted as it is only present in one tree, maps from (d) and (e) are identical, and the not counted associates are the same in both cases.

secondary BPA, violate this rule" (WB, p. 345/760). This claim, however, is false. Under Page's methods (Tree reconciliation and Tree Mapping) when two adjacent terminals do not form a monophyletic group and have the same associate, they are counted as the product of a duplication with an extinction, or as a dispersion (Brooks, 2003, p. 444, erroneously thinks that duplication can be the only explanation). Both these events have the same effect in the reconciled trees: several instances where the same associate(s) is (are) found successively (see Fig. 1).

In conclusion, TM-MC and PACT have a similar basic procedure for merging nodes. Apparent differences in handling the associate lists are just different ways to implement the Tree Mapping (Fig. 2). In fact, it is possible to generalize the best matching for a node for both the TM-MC/PACT algorithm: the best image of a node from tree A in tree B is the node that produces the smallest set that maximizes the number of shared elements (i.e., shared associates) and minimizes the number of discarded elements (i.e., associates absent in one of the trees, or hypothesized to be acquired by associate-switching) (see Fig. 2).

[3]Note that the WB's rule "Y + N = YN" is in fact "Y + YN = YN", or any match could be justified.

Equivalence of optimality criteria

By starting with a tree and adding the other trees one by one, WB presented PACT as an algorithm that never discards a previous move (a greedy[4] algorithm). In contrast, Page's TM-MC algorithm uses an explicit optimality criterion (i.e., the number of codivergences). Page noted that allowing associate-switching and duplications will result in several possible reconstructions. He suggested, "a natural criterion for choosing a reconstruction is maximizing the extent of [codivergences]" (Page, 1994b, p. 164). WB were not explicit about how to select among several possible reconstructions, however, they recommended to "[maximize] the matches between their respective leaves and nodes, and then adding novel elements by creating novel nodes at appropriate levels in the template [host/area] cladogram" (WB, p. 343/758). Under this rule it is possible to match all nodes by simply duplicating all elements (as noted by Page, 1994a; for Tree reconciliation). Brooks and Ferrao (2005, p. 1294)[5] suggested, however, a new rule absent in WB's paper: "duplicate only enough to satisfy assumption 0". Therefore, in PACT as in TM-MC, the stated aim clearly is to minimize duplications and thus to maximize the number of codivergence nodes: the matches that are not products of duplication.

Equivalence of results

As we have shown in the comparison of PACT with TM-MC, while PACT's primary objective is to merge trees, TM-MC deals with the mapping of nodes. The results of both procedures look are at first sight different; however, there is a fundamental equivalence between a merge and a map of two trees. When a map between trees is given, it can be used to merge the two trees in a straightforward and unambiguous procedure (see Fig. 3). Actually, Tree reconciliation has been used to merge trees (Page, 1993, 1994a). Hence, PACT's tree merging is not a relevant difference with TM-MC, as the derivation of a map between trees is the key point of both methods. So far, TM-MC and PACT have similar basic procedure and optimality criterion.

Wojcicki and Brooks (2004, pp. 359–360/768–770) claimed that the use of PACT to compare associate phylogenies with TACs, can "mimic a priori [Tree reconciliation, Tree Mapping] methods of analysis" (Wojcicki and Brooks, 2004, p. 359/768). WB suggested the use of PACT for comparing trees, rather than fusing

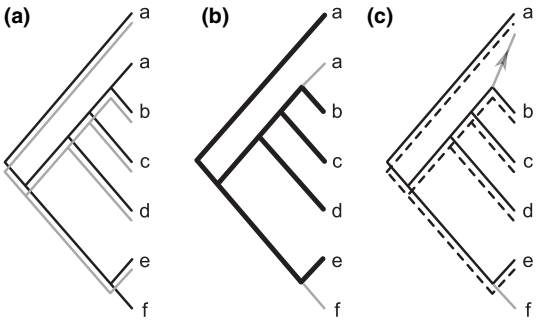

Fig. 3. An example showing how the fusion is done after the map is known. The trees to be fused are trees (a) and (b) from Fig. 2. (a) trees superimposed based on the map; (b) PACT answer, the new elements are added (marked in gray), thick lines denote coincidence between the two trees; (c) the reconstruction showing host dispersals as is usual under TM-MC reconstructions.

it. The results in either case will be the same, since as we have shown so far, PACT is not a mimic, it is a TM-MC method. Under TM-MC incongruent elements are expressed as host switches (Page, 1994b), then it is false that "PACT is the only a priori method that produces direct representations of host reticulations in coevolution" (Wojcicki and Brooks, 2004, p. 360) as TM-MC was explicitly developed to do this a decade before.

A difference between PACT and TM-MC

There is a feature of PACT that makes it different from TM-MC. When a node in the input tree has an image on the template tree, but none of its non-terminal descendants have an image on the template tree, the nodes are left uncombined, similar to a duplication event. For example: (A(B(CD))) and (D(C(BA))) will produce ((A(B(CD)))(D(C(BA)))). TM-MC solves it by proposing multiple associate switches (i.e., PACT's answer shown above) as no codivergence event is found, or proposing several duplications (as in tree reconciliation) and all answers in between. This feature, however, is found neither in WB's table 2, nor in the description of the rules, but only mentioned in a small example (Wojcicki and Brooks, 2004, pp. 345–346/761–762). In either case, two trees having the same elements and contradicting completely one to another seems like a rare event in real data sets.

Based on the above comparison we could not find substantial evidence supporting PACT as a new approach, and therefore were consider PACT an extension of TM-MC.

Problems with PACT's original description

Some particular points about PACT are difficult to understand given WB explanations. WB's rules are often

[4]A particular kind of algorithm, which follows a metaheuristic way to solve the problem. It finds a local optimum at each stage with the hope of finding the global optimum.

[5]Here we do not discuss assumption 0, we only remark that the meaning of this phrase is clearly to minimize the number of duplications.

ambiguous and difficult to apply objectively. Here we single out a number of problems.

Grouping on nodes or single associates?

WB claimed (pp. 345–446, figs 5 and 6/p. 761, figs 11 and 12) that PACT produces groupings based only on nodes: "single hosts [= associates] are not sufficient grounds for grouping or combining [associates]. PACT will not create groupings of hosts [=associates] in the absence of any evidence of groupings" (p. 345/760). However, WB's examples disagree with this claim. In their example 1 (p. 344/759), when combining (CD) with (C(DA)) WB said that "C is the common element in both cladograms, and can be combined. This leaves D … and (DA) … connected at the same node. This means that both D's can be combined'. Note that C is used *first* (only one leaf) and then, *after* that, it is concluded that D can be combined; in these examples a group is based on a single associate, a practice otherwise explicitly prohibited by WB. Moreover, in example 1 (p. 344, fig. 4/pp.759–760, fig. 10) as well, WB combined (A(BE)) with (A(CD)) as (A((BE)(CD))), so they combined (BE) and (CD) based on a single associate A. A similar type of reasoning appears to be followed in their example 2 (pp. 346–348/761–763) where a node (BE) is to be matched. WB refused to combine it with "B" in the template tree because "[it] is a leaf grouped with a node", on the contrary they prefer (CE) because "E is a leaf grouped with leaf C". The essential part of the argument presented here appears to be based on a single shared element, E.

Arbitrary combinations

Some other combinations of trees are difficult to understand following the algorithm and appear to be arbitrary. In WB's third example (pp. 348–351/763–766), the first two cladograms are different in the node (R(E(AL))) and node (T(R(EE))), R and E are the shared associates between the two nodes, and there are not contradictions between them. So the combination must be (T(R((EE)(AL)))), with RE + AL and T + REAL as cases of "Y + YN = YN". Instead, WB chose not to combine the elements ((R(E(AL))) (T(R(EE)))) because "PACT cannot determine if either or both are the same as E + R" (WB, p. 349/763). This result is not implied by any of PACT rules, and contradicts the aim of the algorithm of maximize matches; this is because any combination could be prevented arguing that it is unknown if shared elements are the same. It seems like the decision of excluding TREAL, was based on WB's desire to put the word "REALTREE" in the final answer.

Missing algorithm to detect incongruence

Both PACT and TM-MC provide a strategy to deal with an incongruent element: remove it from the associate set, and add it later as a new element. In manageable and simple examples, it is possible to detect the incongruent element(s) just by eye, but under more complex situations (or under a computer implementation) the identity or the criteria to deal with elements susceptible to be removed has to be established objectively.

WB never explained how the user of their algorithm could guess and evaluate associates removal, then any choice will be as arbitrary as SBPA duplications (see Siddall, 2005). Page's (1994b) description of TM-MC provides an exact and heuristic method based on a trial-and-error approach to detect nodes that could be removed from the associate set. Moreover the eliminations are justified as they represent a net increase of matching nodes between the cladograms compared (the optimality criteria).

Ambiguity and overcombining

Given that PACT is described as a downward algorithm, the matching of nodes stops as soon as a matching node is found. Given two trees (X(A(BC))) and (Z(Y(A(BC)))) that will be fused using PACT, the node (A(BC)) in the first tree matches (A(BC)) in the second tree. More basal nodes on the second tree would match perfectly (A(BC)) but they do not increase the number of shared elements. The problem is the node (X(A(BC))), it would be basal to the (A(BC)) in the second tree, but its position with respect to Z and Y is ambiguous. Given the examples from WB, it seems PACT algorithm halts in the first possible node to proceed with the fusion, which result will be (Z(XY(A(BC)))). According to WB, this solution is the best one: "when there is ambiguity in phylogenetic data, we have greater confidence in the oldest rather than the most recent data. (…) While this result is a possibility, there is no direct evidence supporting such an *a priori* rooting. (…) so PACT produces a more conservative, and *preferable*, result" (WB, p. 348; their italics, our underlining). The answer of PACT is not more conservative. Furthermore the same argument used for (Z(XY… could be used to defend (ZX(Y… after all, we do not have direct evidence for either grouping.

WB's words imply that taking always the first possible node it is the conservative and preferable solution. This move hides that there is ambiguity in the data because the "conservative" answer is a stopping rule for the algorithm. It would be, however, preferable to acknowledge the problem, and offer some solutions, or at least alert the users of the algorithm about this feature.

In the example presented here there is no way to resolve the ambiguity: nodes can not be collapsed

without producing a contradictory answer: (ZXY… is contradicted by second tree. A possible solution could be to use a symbol indicating that the placement of X is ambiguous, and try a better placement (i.e., unambiguous) if an extra clade is added, or if no better placement is found and any new placement is costly, leave the answer marked. This problem is also present in TM-MC and it is important that WB point it out, but it is also deceiving that they try to convince the reader that an algorithm's shortcoming is a preferable and more conservative answer.

Dependence on input order

According to WB, the order in which the TACs are included does not change the final answer (WB, table 2/2), but they do not explain why or how this conclusion is reached. If PACT is executed as described by WB a change in the input order does have an impact on the final answer. Using WB's example 1, and the suggestion in table 2, we chose a complex cladogram as an initial template, in this case their number 4 ((A(B(CD))) (A(B(CD)))). Adding cladograms 1–3 is not a problem, when adding cladogram 5, the new topology is ((A(B(CD)))(A((BE)(CD)))). Then adding cladogram 6 the two solutions are: ((A(B(C(DA))))(A((BE)(CD)))) or ((A(B(CD)))(A((BE)(C(DA))))). Those solutions are maintained with the addition of cladograms 7–9, and both solutions are different from WB's solution: (A(A((BE)(C(DA))))), which they arrive after starting from cladogram 1.

Failure to find optimal solutions

The greedy algorithm of PACT does not always find optimal solutions in more complex problems. For example, if the template tree is ((G(AB))(C(D(E(-G(AF)))))), and the input cladogram is (C(D(E(-G(AB))))), PACT immediately finds that (AB) are equal, so fuses both; likewise (G(AB)) is also present, but C D E are absent in this part of the tree and must be added later than (G(AB)). The new template cladogram is ((C(D(E(G(AB)))))(C(D(E(G(AF)))))), and the solution produces two matches. A better solution is possible, when B is allowed as a new element (an associate-switch), and then the solution is ((G(AB))(C(D(E(G(-F(AB))))))), which matches all but one node of the input cladogram, and only one element is duplicated, so it is preferable under Brooks' proposed optimization criteria (maximization of matches, minimization of duplications). Of course, all heuristic solutions, including the one proposed by Page (1994b), can produce non-optimal solutions. However, heuristics using global examinations tend to find solutions closer to the optimal than any greedy solution, greedy solutions could be used as a starting point.

Interpreting PACT answers

Equivalence of PACT results and tree maps

PACT results are neither general associate cladograms (GAC), nor reticulograms. When we look for a GAC, the result would be a representation of associates in a unique hierarchical fashion without repeated elements. A GAC taken "as is" might be contradictory with some TACs (as a cladogram is contradictory with the homoplastic characters), but extending them to include duplications and host shifts in the representation produces fully congruent trees (see Fig. 3). This is the form in which PACT answers are presented. PACT trees are "extended" trees, analogous to Page's (1994a) M-trees. Although showing reticulation, the extended answers are not reticulograms. In a reticulogram there are several possible hierarchical relationships (product of the multiple ancestors), with explicit loops among edges (nodes and terminals). Furthermore in a reticulogram terminals are not repeated. It is possible to convert the PACT's extended answers in a reticulogram, although this could be a difficult task without knowing the associations at each node.

Identification of common elements

As a trade-off for avoiding the "ambiguity by over combining data" (WB, p. 345/760), the trees presented by PACT do not indicate common patterns to multiple input trees, requiring the researcher to recheck (manually) the data to find any common patterns [equivalent to the a posteriori interpretation of Wiley (1988)]. WB's example 3 shows this, the group (IS) is in 14 of 17 input cladograms, and in 11 of them is the sister group of a clade that contains T [usually as (TH)]. Yet in the final result this relationship is completely lost. In fact, the final answer contains two (IS) associated with (TH): the group ((TH)(IS)) and ((IS)((T(HE))(.))), but given that the IS group was duplicated, the relationship of IS, and (TH)(IS) is lost without examining the original trees. The problem is a by-product of dismisses the search for a GAC. Then, new elements, even if they are already present in the solution, are always added to the answer, but never collapsed into a single element. Therefore, adding new cladograms will make the solution larger and confusing.

WB claimed that "PACT is an excellent tool of discovery"; however, without an explicit method of reconstruction of the taxon histories based on the PACT-tree, it is not clear what is being discovered. Two techniques are presented to understand the expanded tree answer of PACT. The first one was offered by WB and is referred here as the "window method", where groups of terminals are enclosed in sets of windows (WB, pp. 353–359, fig. 44/p. 768–773, fig. 50; WB did not give an

explicit name to this technique). WB explained that each window represents a different time frame. The history of the elements of each window is described without taking into account elements in the more exclusive window. WB described the process of building windows as a heuristic tool to understand the results, but do not indicate how these windows should be built. This looks like a subjective heuristic (because there are no rules to construct the windows). The windows described in the text (WB, pp. 353–358/768–773) are different from the windows in the figures (WB, fig. 44/50; our Fig. 4). We might think that this is a typographical error, but it is unlikely that the same error was unnoticed in both WB papers. A second option is that WB designed the windows to show that they could be changed, without mentioning so in the text. The "heuristics" of the window method appears to be an arbitrary selection of one description over many other equally possible descriptions.

In another paper a new technique is presented instead of the window method. We call this technique "associate tracing", because it underlines the associate phylogeny

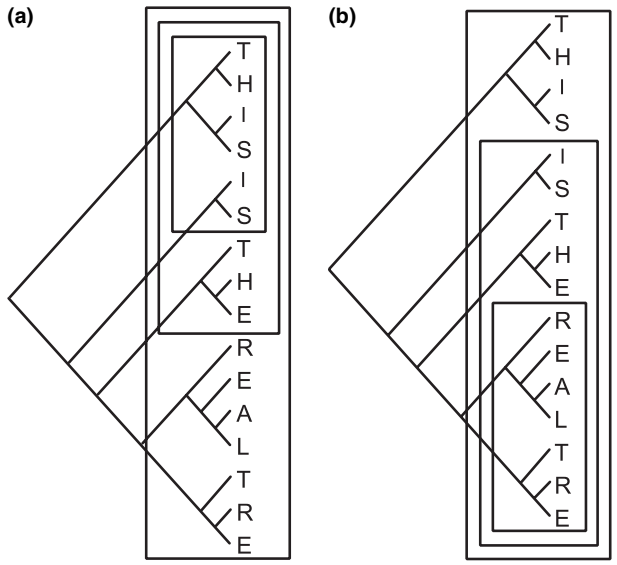

Fig. 4. Described windows in WB paper: (a) in the text the description states, "three rectangles heuristically depicting increasing temporal scale. In this case, the shortest temporal scale (smallest rectangle) encompasses only four hosts, T, H, I and S, two of which (I and S) exhibit reticulated relationships. As we expand the temporal scale (medium rectangle), we add an additional host (E), but also add reticulated relationships for hosts T and H. Finally, at the longest temporal scale (largest rectangle), we add three additional hosts, R, A and L, and also additional host reticulations for hosts E and R." WB (pp. 354–358/767); (b) the printed figure and its caption: "rectangles representing heuristic view of increasing temporal scale associated with increasing host range and reticulated host relationships. The smallest rectangle indicates five hosts and two reticulations, the medium-sized rectangle indicates eight hosts and three reticulations, and the largest rectangle indicates eight hosts and eight reticulations." (WB, fig. 44/50).

in the expanded answer obtained using PACT (Brooks and Ferrao, 2005; they did not propose an explicit name for this method). The non-underlined lineages are taken as associate-switches, dispersals, or duplications according to the circumstances. The technique is strikingly similar to no-technique-at-all (i.e., subjective association), which predominated in the early host–parasite association studies (see, for instance, similar pictures in Hennig, 1966, figs 30–32, 56 and 57). The problems with associate tracing are that

1 if you use this approach, then there is no need for an analysis, we could just trace the associate in each parasite independently and reach the same conclusion. In fact the "results" of associate tracing used by Brooks and Ferrao (2005) are identical using or not using PACT (i.e., with the original TACs): all events of incongruence between associates and parasites are found, and so are their biogeographical implications, and

2 there are no rules for reconstruction. Several tracings might be possible (and this was the reason to develop BPA, TM–MC and other association methods); therefore one requires a rule of thumb to resolve incongruence between the associate and host phylogenies (note that Brooks explicitly defends that the phylogenies of the parasite and the associate do not need to be congruent (e.g., Brooks, 2003).

The method of "host tracing" does not represent progress, but a return to a time when host–parasite associations were studied without an explicit optimality criterion.

Orthogenesis?

In many papers, Brooks and collaborators (e.g., Brooks, 2003; Brooks and Ferrao, 2005, p. 1292) equate maximum codivergence methods with Tree reconciliation and reject it because it explains incongruence using duplications and extinctions, and then every possible host-switch instance can disappear from the final answer. This is a misrepresentation of maximum codivergence methods: Page was fully aware that Tree reconciliation prohibited associate-switches (Page, 1994a,b) this is why he developed TM-MC (Page, 1994b). As a product of Brooks' misunderstanding of TM-MC, he considered it as "orthogenetic" or based on models (Brooks, 2003; Brooks et al., 2004), whereas he advocates that his own methods (BPA, SBPA and PACT) are "phylogenetic systematics used in the context of [associates] and parasites" (Brooks, 2003, p. 444). In this section we show that some of the claims of Brooks about Tree Mapping are without foundation.

According to Brooks and Ferrao (2005) the orthogenetic program relies on a model of coevolution that implies ecologically specialized parasites. However, the only assumption of TM-MC in this context is that

organisms of compared trees share, at least partially, a common history, which is the same assumption that is implicit under PACT, BPA and other historical association methods. Using Brooks' analogy between phylogenetics and associations we can say that TM-MC minimizes *ad hoc* hypothesis of "lineage" homoplasies, which means to minimize the number of what cannot be explained as matches (codivergences), hence to maximize codivergences. But this does not imply that such lineage homoplasies are rare; just as in phylogenetic reconstruction, parsimony does not imply rarity of homoplasy (Farris, 1983).

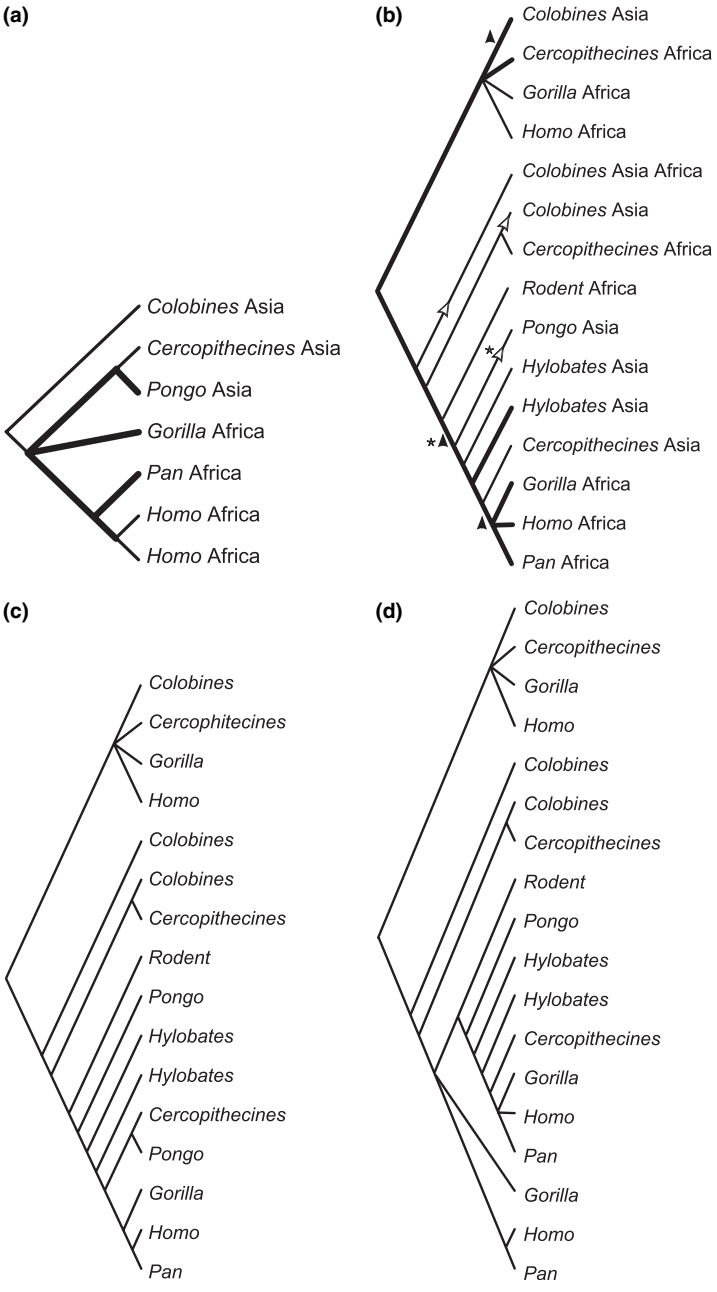

Fig. 5. PACT and TM-MC analyses of *Enterobius* and *Oesophagostomum*. (a) Host–parasite cladogram of *Enterobius*, showing distribution of parasite-host assemblages, thick lines as in Brooks and Ferrao (2005) showing host (Primates) phylogeny. (b) Host–parasite cladogram of *Oesophagostomum*, showing distribution of parasite-host assemblages, thick lines as in Brooks and Ferrao (2005) showing host (Primates) phylogeny; arrows show host-shifts coupled with range expansion of parasite, filled triangles show congruent pattern between host and parasites phylogeny, but treated by Brooks and Ferrao as "host shifts" because of the range expansion of the parasite; asterisks show two possible alternatives to optimize infestation in *Pongo*. (c) PACT result as shown by Brooks and Ferrao (2005); (d) answer found here after expanding (*Colobines*,(*Hylobates, Rodent, Gorilla* (*Pan, Homo*) (*Cercopithecines, Pongo*))), the TM-MC result.

In any analysis of association (gene–species, parasite–host, species–area), the first objective is to find the *common history* between the studied organisms. It is for this reason that Page (1994a,b) and, 10 years later, WB and Brooks and Ferrao (2005) maximize matches between independent phylogenies. When common events (as codivergences) or independent events (as associate-switches) are found, it is possible to reconstruct the history of the association optimizing those common events. Without knowing the common history of primates and parasitic worms, Brooks and Ferrao would never have found any answer to their question (the number of host-shifts). Brooks and Ferrao's study shows how necessary it is to compare host and parasite phylogenies in order to discover something; using PACT without some sort of reconstruction (or even with an ambiguous tracing) is completely useless.

The empirical example of Brooks and Ferrao (2005, p. 1295) "clearly support ecological fitting more than maximum co-speciation". But it is difficult to know how clear this support is, because Brooks and Ferrao (2005) did not perform any comparison! Our reanalysis of their data, using simple parasite-host tracing as well as the TM-MC analysis (Fig. 5) found the same general conclusions as Brooks and Ferrao. In fact, as expected with "ecological fitting" but not with orthogenesis, we did not find a close association between the associate and parasite phylogenies (under the classic TM-MC answer, a fully congruent "tracing" is not possible). Like Brooks and Ferrao, we did not find evidence to discriminate host switching or sympatric speciation (duplications) for *Hylobates'* parasites. Moreover, under our analysis, dispersal from Asiatic *Oesophagostomum* to African great apes also fits the emergence of infectious diseases with periods of range expansion. All other dispersals "discovered" by Brooks and Ferrao were on leaves, so they are independent of the reconstruction.

A special point remains unanswered in Brooks and Ferrao (2005): why does the application of TM-MC make a researcher an "orthogeneticist", whereas tracing a host using PACT does not?

Concluding remarks

The assertion that PACT is the first method that "escaped from the matrix" and dealt with "associate reticulation" is false. These particularities are already present in the first TM-MC description (Page, 1994b). PACT is more a modified TM-MC algorithm than a new one. The development of PACT shows that the gap between different methods is not as wide as Brooks believes (Brooks, 2003; Brooks et al., 2004; Brooks and Ferrao, 2005; see also Stevens, 2004).

There is not yet fully developed software that runs under PACT's algorithm. The interpretation of TM-MC's results using the current software requires doing some hand-work (to do the maps, and move across programs) and requires using fully dichotomous cladograms. Nevertheless, as PACT and TM-MC use the same approach, hopefully a more powerful approach will come from either of the two sides.

Acknowledgments

We would like to acknowledge the many authors that leave their papers available on line. D. R. Brooks provided us with a beta version of his implementation and read an early version of the manuscript. M. Ebach and M. Quijano helped with constructive discussions with early drafts. Two anonymous referees helped to make our initial perspective clear and one of them was extremely helpful to improve our final version. Jessica Hawk and Kimberly Terrel helped us with the English grammar and writing style. The financial support of the Vicerrectoría de Investigaciones y Extensión—Universidad Industrial de Santander, is kindly acknowledged (Project 5132).

References

Brooks, D.R., 1981. Hennig's parasitological method: a proposed solution. Syst. Zool. 30, 229–249.

Brooks, D.R., 1990. Parsimony analysis in historical biogeography and coevolution: methodological and theoretical update. Syst. Zool. 39, 14–30.

Brooks, D.R., 2003. The new orthogenesis. Cladistics 19, 443–448.

Brooks, D.R., Ferrao, A.I., 2005. The historical biogeography of co-evolution: emerging infectious diseases are evolutionary accidents waiting to happen. J. Biogeogr. 32, 1291–1299.

Brooks, D.R., Van Veller, M.G.P., McLennan, D.A., 2001. How to do BPA, really. J. Biogeogr. 28, 345–358.

Brooks, D.R., Dowling, A.P.G., Van Veller, M.G.P., Hoberg, E.P., 2004. Ending a decade of deception: a valiant failure, a not-so-valiant failure, and a success history. Cladistics 20, 32–46.

Farris, J.S., 1983. The logical basis of phylogenetic analysis. In: Platnick, N.I., Funk, V. (Eds.), Advances in Cladistics, Vol. 2. University of Columbia, New York, pp. 7–36.

Goodman, M., Czelusniak, J., Moore, G.W., Romero-Herrera, A.E., Matsuda, G., 1979. Fitting the gene lineage into its species lineage: a parsimony strategy illustrated by cladograms constructed from globin sequences. Syst. Zool. 28, 132–168.

Hennig, W., 1966. Phylogenetic Systematics. University of Illinois Press, Chicago, IL.

Page, R.D.M., 1993. Component, version 2.0. Natural History Museum, London.

Page, R.D.M., 1994a. Maps between trees and cladistic analysis of historical associations among genes, organisms and areas. Syst. Biol. 43, 58–77.

Page, R.D.M., 1994b. Parallel phylogenies: reconstructing the history of host parasite assemblages. Cladistics 10, 155–173.

Siddall, M.E., 2005. Bracing for another decade of deception: the promise of Secondary Brooks Parsimony Analysis. Cladistics 21, 90–99.

Stevens, J., 2004. Computational aspects of host–parasite phylogenies. Brief. Bioinf. 5, 339–349.
Wiley, E.O., 1988. Parsimony analysis and vicariance biogeography. Syst. Zool. 37, 271–290.
Wojcicki, M., Brooks, D.R., 2004. Escaping the matrix: a new algorithm for phylogenetic comparative studies of co-evolution. Cladistics 20, 341–361.
Wojcicki, M., Brooks, D.R., 2005. PACT: an efficient and powerful algorithm for generating area cladograms. J. Biogeogr. 32, 755–774.

Appendix 1. Formalization of a TM-MC/PACT heuristic

The formalization is presented as pseudo-code following, loosely, the C syntax.

Problem

At least two trees (tree1 and tree2) of historical associations to be merged.

Definitions

An Associate Set is a list of associates (areas or hosts) for a particular terminal or node.

A taxon-by-association tree (TAC) is a phylogenetic tree where each terminal is replaced by its set of associates (i.e., areas or hosts).

A match between two nodes indicates that both associate sets are identical.

A map is the list of matched nodes.

Let a *s*-node to be a node in which at least two of its descendants are *s*- or *j*-elements (terminals or nodes). At the beginning of the algorithm, all elements are *s*-elements. Let an *l*-element an incongruent element, *l*-elements are treated as new elements. Let a *j*-node to be a node in which, as much, only one descendant is an *s*- or *j*-element, and the other descendants are *l*-elements, *j*-elements are treated as new elements.

*Procedure assign_associate_sets (TAC **MyTree**)*

Using a post-order traverse across ***MyTree*** nodes, for each node of ***MyTree***, let the associate set void, then add all associate sets of its descendants except if the descendant is an *l*-element.

*Function make_map (TAC **input_tree**, TAC **template_tree**, map **MyMap**)*

For each *s*-node of ***input_tree***, find its image among *s*-nodes of ***template_tree***.
The image of a node of ***input_tree*** is the node of ***template_tree*** that maximizes the number of shared associates (i.e., maximizes the cardinality of both associate sets intersection). If there is a tie the image is the node, among the previous candidate nodes, that minimizes the number of non-shared associates. If the tie persists, the image is among the previous candidate nodes, the most derived node, if candidates are in the same clade. Otherwise an arbitrary node, among candidates, is selected (*).
End for
For each *s*-node of ***input_tree***, find whether its image is also a match.
Let the number of matches as ***k***.
Store the matched nodes in ***MyMap***.
An image is a match if the node of ***template_tree*** is not image of any other node of the ***input_tree***. If a node of ***template_tree*** is image of several ***input_tree*** nodes, then only the most derived node (if all nodes pertains to the same clade) of ***input_tree*** is counted as matched, otherwise and arbitrary node, among images, is selected as matched (*).
End for
Return ***k***

*Function Find_the_map (TAC **input_tree**, TAC **template_tree**)*

The reference TAC is ***template_tree***, and the second TAC is ***input_tree***.
Leave all-non-shared terminals as *l*-terminals, and shared terminals as *s*-terminals.
assign_associate_sets (***input_tree***)
assign_associate_sets (***template_tree***)
Let the set of *l*- and *j*-nodes of input tree as ***J***. Let ***d*** = (cardinality of ***J***), the number of unmatched nodes of the input tree. Let ***K*** = 0, the maximum number of matched nodes. Let the set of matched nodes as ***M***, and ***N*** the actual set of matched nodes.
actual_k = make_map (input_tree, template_tree, N)
M = ***N***
If (***actual_k*** > ***K***) then ***K*** = ***actual_k***
If (***K*** = *s*-nodes_of_input_tree) then return ***M***. The maximum number of matches is found.
For each *s*-terminal in input_tree = ***myTerminal***
Turn ***myTerminal*** to a *l*-terminal assign_associate_sets (***input_tree***)
actual_k = make_map (***input_tree***, ***template tree***, ***N***)
If (***actual_k*** > ***K***) then
K = ***actual_k***
M = ***N***
add new *l*- and *j*-nodes to ***J***, and let ***d*** = (cardinality of ***J***)
reset and restart the loop
Else ***myTerminal*** as *s*-terminal
End if
If (***d*** = ***K*** − 1 − *s*-nodes_of_input tree) then return ***M***
End for
Return ***M***

Procedure Combine_trees (TAC ***input_tree***, TAC ***template_tree***, map ***M***)

Here an added node is a node of ***input_tree*** which is already combined with ***template_tree***, *s*-terminals count as added elements.

For each node of ***input_tree*** in ***M*** = *i*-node

Let t-node as *i*-node image in template tree.

If all non-terminals descendants of *i*-node are not also ***M*** nodes then create a new node in the ***template_tree***, *u*-node, in which descendants were the *i*-node and the *t*-node, and its ancestor is the original ancestor of *t*-node, the *u*-node is the new image (and also the new match) of the *i*-node.

Else both nodes are equivalent, add new terminals (i.e., *j*-terminals, and terminals absent in t-node associate set) of *i*-node to *t*-node, if a non-shared descendant node of *i*-node, *x*-node, does not have shared descendants (i.e., no descendant of *x*-node is in ***M***, or is not a *s*-terminal), add *x*-node to *t*-node.

End if

All elements of *i*-node are added nodes (‡)

End for

For each non-added nodes of ***input_tree*** = *i*-node, in a post-order travel

If a descendant of *i*-node is already added, as a *t*-node in the ***template_tree***, and the image of ancestor of *t*-node is in ***M*** (i.e., is a matched node), or is the root of template tree, create a new node in the ***template_tree***, *u*-node, in which descendants were the *i*-node and *t*-node, and, if it is not the root, its ancestor is the already added *t*-node ancestor.

Else if a descendant of *i*-node is already added, as a *t*-node in the ***template_tree***, but the ancestor of *t*-node has no image in ***M***, add new elements of *i*-node to *t*-node.

Else if a descendant of *i*-node is already added, as a *t*-node in the ***template_tree***, and *t*-node is the root of ***template_tree***.

End if

In all three cases non-shared terminals of *i*-node (*j*-terminals, terminals absent in *t*-node associate set) and non-shared nodes are added to ***template_tree*** at *i*-node (§).

End for

If the basal clade of the ***input_tree*** remains not added to ***template_tree***, add it to the root of template tree.

End procedure.

Main (TAC_list ***tree_pool***)

The main function of the algorithm

Let first tree in ***tree_pool*** as ***template_tree***

For each tree in ***tree_pool*** = ***input_tree***

If ***input_tree*** different from ***template_tree***

Let ***M*** as the list of matched nodes

M = Find_the_map (***input_tree***, ***template_tree***)

Combine_trees (***input_tree***, ***template_tree***, ***M***)

End if

End for

Store ***template_tree***

The description of a generalized TM-MC/PACT is based principally on Page (1994b) TM-MC heuristic. As the objective is to combine trees, instead of turning nodes in *j*-nodes, the terminals were changed to *j*-terminals, then an explicit list of *j*-terminals is always kept. There is no guarantee of founding all optimal reconstructions, and in ambiguous cases, only one solution is kept, see under (§) below.

The algorithm maximizes the number of shared elements between associate sets. It is possible, specially in simple cases, in which turning a *s*-terminal into a *j*-terminal does not change the number of matches, that in the procedure Combine_trees not only *j*-terminals are added, but also, terminals not present in the associate set of the node in the template tree. For example the template tree is (a(b(cd))) and the input tree is (a(b(c(ad)))), the algorithm found three matches, if the a in (ad) is changed to *j*-terminal, the number of matches does not change, so it is restored as *s*-terminal at the end of the loop in Find_the_map. A second option, is that if a match is found and there are some non-shared terminals, these elements would be forced to be *j*-terminals, and the associate set of their ancestors would be changed in consequence.

The present algorithm shares with original PACT description all WB's rules (YN + YN = YNN and its particulars, and Y(Y– = Y(Y–, if Y is a terminal or a clade), attempts to maximize the number of matched nodes, and incorporates an heuristic to duplicate incongruent elements, so it is legitimately a formalization of PACT algorithm.

Under our definition of match, not only nodes product of dispersal would be duplicated (i.e., added), but also nodes from a strict duplication (which are not allowed by WB, see text under "Interpreting PACT answers"); to evade several duplication, it is possible to force a comparison between nodes and clades that were matched in a previous "weighted" clade. Considering for example the template tree (a(b(cd))), the first input tree is (a(b(c(de)))), then the new template tree is (a(b(c(de)))), a second input tree is ((a(b(cd)))(a(b(cd)))), then the new template tree is ((a(b(cd)))(a(b(c(de))))), in which (abcde) clade is the "weighted" clade, so adding (a((bf)(cd))) and (a(b(c(da)))) would produce ((a(b(cd)))(a((bf)(c(e(ad)))))), all new elements being added to the "weighted" clade.

Also, it is possible to compare a tree with itself, if nodes compared are placed in independent clades. Considering for example (a((b(cd))(e(fg)))) (cd) could be compared with a, b, e, f, g (if some of them are clades) (e(fg)), and (fg), but not with c, d (if some of them are

clades) (b(cd)) ((b(cd))(e(fg))), or the whole cladogram. In this case, only a map could be produced without making a combination. This could be done in the final part of the analysis for the template tree to find common patters, pulled out by the expanded answer (see text under "Interpreting PACT answers").

We also remark a problem with polytomies, that although they could be solved for simple cases, they are more difficult in complex ones. For example, combining (abcd) with (a(b(cd))) produces (acd(b(cd))). A solution could be to link directly *s*-terminals of input tree with the terminals of template tree after the map search, but this is problematic in cases in which terminals are shared, but nodes are contradictory. Moreover, resolving the clade depends on the first tree that resolves the clade. In the example above it would be (a(b(cd))), but if another tree with clade (d(c(ab))) is added then the WB option enters in operation. Under the present algorithm, the first union is as previously given (acd(b(cd))), while the third tree is (ac(b(cd))(d(c(ab)))).

Notes

(§) Under the present algorithm, ambiguous solutions by-product of non-shared leaves produce always the same answer, even if there is no evidence for a particular grouping. Considering the example (a(b(c(d(e(fg)))))) and (a(h(i(e(fg))))), there are several possible solutions for nodes b-,c-,d-,h- and *i*-. The present algorithm always prefers (a(b(ch(di(e(fg)))))). It is worth noting that WB did not provide any clue about a possible solution, and as Page only dealt with maps with fully common associate sets, that was not a concern in his algorithm.

(*) It is possible to speed up, and also avoid contradictory matches, by restricting the search for an image to the ancestors of the clade in the template tree in which is found an image to a descendant of the *i*-node. In this case the algorithm could be fooled in the same way as the original WB's PACT description (see text under "Problems with PACT's original description'). In Find_the_map main loop the associate set is perturbed, so in some cases it is possible to overcome the fool answer.

(‡) This step takes into account the novel characteristic of WB's description (see text under "A difference between PACT and TM-MC"), of-non-combining when only the host set is shared, but not the association, as in (a(b(cd))) and (d(c(ab))), the WB's example (WB, pp. 345–346/760–761). Under our definition the situation also covers the case with three terminals: (a(bc)) and (c(ab)). This option could be "turned off" ignoring the first case, in which the union would be (ad(b(cd))(c(ab))) or (ac(bc)(ab)) for each case, respectively.