A note on Brazeau et al.'s (2017) algorithm for characters with inapplicable data, illustrated with an analysis of their Fig. 3d using anagallis, a program for parsimony analysis of character hierarchies.

Jan De Laet, Göteborgs Botaniska Trädgård
Original version November 4, 2017
Minor corrections November 5, 2017

Keywords: parsimony analysis, homology, inapplicable data, character hierarchies, character optimization.

Abstract - Brazeau et al. (2017) recently published a paper with a single-character algorithm to calculate the score of a character with inapplicable data on a tree, aiming to maximize homology in such characters. In this note, I show by example (using their Fig. 3d) that their algorithm is insufficient to find all optimal inner node state reconstructions in such characters. The root cause seems to be a inherent built-in constraint on the evaluation of implied absence/presence characters in their algorithm. In more complex cases, this constraint can lead to an overestimation of the optimal minimal score of character hierarchies and to errors in the trees that are selected as optimal during tree search.

Content

Introduction: maximization of homology as a solution to the problems with inapplicables

In parsimony analysis, the problem of inapplicables (Maddison 1993) can be overcome by maximizing the amount of similarity that can be interpreted as homology, an idea that I first discussed in a talk at the 2002 meeting of the Willi Hennig Society in Helsinki (De Laet 2002). In this, similarity is used in its technical meaning: an observed shared similarity amounts to a prior hypothesis of homology that is rooted in empirical observation. Some more background can be found in De Laet (2012a, 2012b, 2013a, 2013b). A recent discussion can be found in the section on inapplicables of De Laet (2015).

Maximization of homology also provides the key to extend parsimony to the analysis of unaligned sequence data (De Laet 2003a, 2005; see also De Laet and Castraviejo-Fisher 2016). In this context, it can be shown that in tree alignment programs such as POY, cost regime 3221 (gap opening cost three, transition and transversion costs two, and gap extension cost one) provides an optimal approximation for the cost set that maximizes homology when all instances of homology are equally weighted. A discussion of differential weighting of homologies can be found in De Laet (2015, sections on approximations and on sensitivity analysis).

Inapplicables as they arise in the classic approach are a special case of inapplicables as they arise in sequence data. This special case can be tackled with algorithms that are computationally less complex. In brief, to maximize homology accross a full-blown absence/presence character hierarchy, it suffices to minimize the sum of gains/losses, transformations, and regions of applicability (subcharacters) over the constituent characters of that character hierarchy, subject to the constraint that the overall explanation must be free of internal contradictions. Anagallis is a computer program that provides tree searches with such algorithms.

Release of anagallis was originally announced by the 2013 meeting of the Willi Hennig Society in Rostock (De Laet 2013a), but afterwards I decided to postpone release until the program could guarantee optimality of tree scores obtained under a much wider range of conditions than initially announced, which took some more time than I anticipated at the time. In the current development version optimality of reported tree scores is guaranteed as long as no independent regular Fitch or Farris optimization of a particular feature has a solution where an individual region of absence has more than eleven neighbouring regions of presence, a reasonable enough assumption in practice. This unreleased development version is used in this note. A public beta should be available by the end of the year, as announced here on 24 October 2017.

The approach of Brazeau et al. (2017)

Introduction

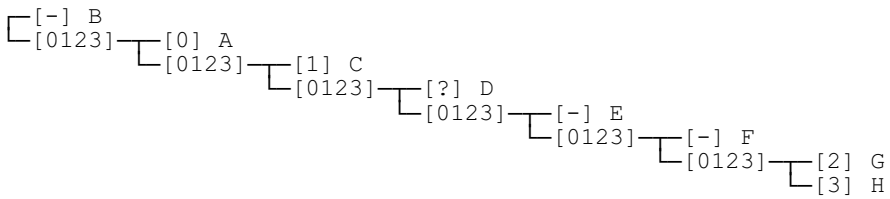
Brazeau et al. (2017) recently published a paper on morphological analysis with inapplicable data on BioRXiv. They explicitly aim to maximize homology in characters with inapplicable data, but the main difference with my approach seems to be that they independently optimize single-column characters with inapplicables rather than character hierarchies as a whole, as done in anagallis. This may give reasonable results under a wide range of conditions, but in general the optimization of a character hierarchy on a tree cannot be reduced to a series of independent single-character optimizations on that tree. Doing so may yield a fast approximation for the score of a character hierarchy on a tree, but it can miss optimal state reconstructions, lead to an overestimation of the optimal minimum hierarchy score, and ultimately lead to errors in trees that are considered optimal. In this note I concentrate on the first of these three issues, using Brazeau et al.'s Fig. 3 as an example.

A detailed analysis of Brazeau et al.'s Fig. 3

Brazeau et al.'s Fig. 3d shows a pectinate tree of 8 unnamed terminals, observed states of some character at the tips, and final optimal statesets at the inner nodes. For convenience, I'll refer to the eight terminals at the tips as A through H. Let's assume that they are bird species. The character that is optimized in that figure is then this one ('-' stands for inapplicability):

```
A   0
B   -
C   1
D   ?
E   -
F   -
G   2
H   3
```

Brazeau et al.'s Fig. 3d can then be represented as follows:



For ease of discussion I'll further refer to it in terms of Maddison's (1993) classic example: character states 0 to 3 are four different tail colors, observed in terminals with tails; and '-' indicates inapplicability of tail color in the terminals that do not have a tail. The optimizations shown in Brazeau et al.'s Fig. 3d are correct, but only part of the story. To get started, let's first explicitly add the character that describes tail absence/presence. Here is one way to do that:

```
A   10
B   0-
C   11
D   ??
E   0-
F   0-
G   12
H   13
```

Here is another:

```
A   10
B   0-
C   11
D   1?
E   0-
F   0-
G   12
H   13
```

The difference is in terminal D. Assume we only know D from a fossil impression. In the first example, it would be an incomplete impression: we don't know if D had a tail because that part has not been preserved in the known fossil record. In the second example, the impression is complete and has been observed to have a tail. We just don't know its color.

But there are still other possibilities. Here's an example. Assume for an instance that terminal B is not a bird but an angiosperm. In that case, the question if B has a tail of the anatomical kind and in the topological correspondences as observed in birds becomes meaningless. In other words, tail absence/presence itself is inapplicable in B. To express that, an additional character is needed:

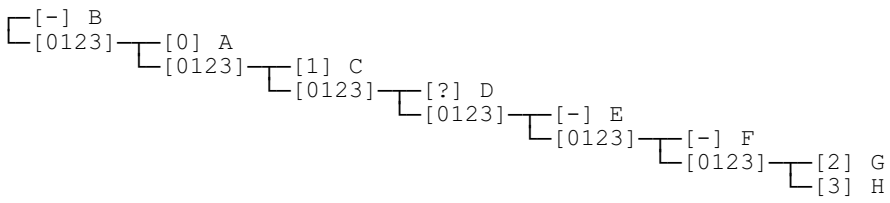
```
A   110
B   0--
C   111
D   11?
E   10-
F   10-
G   112
H   113
```

In the rest of this note, I'll go with the first assumption: D is an incompletely preserved fossil bird, and whether or not it had a tail is missing data (some of the details below would be different with the other assumptions):

```
A   10
B   0-
C   11
D   ??
E   0-
F   0-
G   12
H   13
```

As pointed out by Farris (1983; see also Farris 2008), trees with optimized characters are able to explain shared observed similarities as due to inheritance and common descent. In this dataset, all observed tails have different colors, so there are no observed shared similarities in tail color that need explanation. All observed shared similarities are in the absence/presence of tails: B, E, and F share tail absence, A, C, G and H share tail presence.

Here is Brazeau et al.'s optimization again:



They did not explicitly add a character that describes tail absence/presence, but as all inner nodes are optimized with tail colors, their Fig. 3d implies that the observed tails of A, C, G and H can be explained as due to inheritance and common descent. That amounts to three independent pairwise explained observed similarities (if, for example, shared similarities AC, CG and GH are explained, it follows that AG, AH and CH are explained as well). On the other hand, none of the observed shared tail absences can be explained as due to inheritance and common descent in this optimization. So the grand total of explained observed similarities is three.

But there are still other optimizations with that same grand total, optimizations that Brazeau et al.'s algorithm does not find. For the purpose of this note, I obtained them using the following series of anagallis statements:

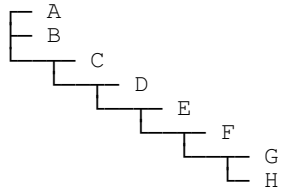
```
characters read numeric
7 8
A 00000 10
B 00000 0?
C 10000 11
D 11000 ??
E 11100 0?
F 11110 0?
G 11111 12
H 11111 13
;
characters set [-/10 1.5  [-/1 6.7;
characters set <6 7>;
tree search mult * 1;
tree show plot b;
characters score;
characters optimize plot bc6;
characters optimize plot bc7;
coppAbc6.7;
```

The first statement, 'characters read numeric', is used to enter the character data. I added the first five characters just to obtain the pectinate tree that is in Brazeau's et al.'s Fig. 3d. Characters six and seven then represent the character hierarchy under discussion. The first 'character set' statement specifies that the first five characters are active, non-additive, and have a prior weight of 10; and that characters six and seven are also active and non-additive, but with a prior weight of 1.

The second 'characters set' statement specifies the character hierarchy, using the default convention that '-' stands for inapplicability. Between '<' and '>' (where nested occurrences of '<' and '>' may be present), the first character is an absence/presence character, the following characters are characters that describe features that are only applicable when that first character has state 'presence'.

The 'tree search' statement specifies a single round of tree building using a random addition sequence and tbr branch swapping. As expected, it yielded a single optimal tree, the same tree as in Brazeau et al.'s Fig. 3d, as verified by statement 'tree show plot' (option b suppresses numbering of the inner nodes):

```
trees search mult> doing 1 replicate with spr swapping
rep 1   score 58   (addition sequence as is)                 [1 tree, best score 58, 1 rep hit 58]
trees show plot> plotting current tree
trees show plot> all leaves included
current tree (1)
```



Statement 'characters score' next gives the scores of all characters on that tree:

```
characters score> current tree (1)
10*1 10*1 10*1 10*1 10*1 8 <c6>
total score 58 in 7 active characters (using non-unity prior weights)

Main character hierarchy at 6
  * root character hierarchy at 6: total score 8
    + 3 gains/losses in 1 subcharacter (score 4)
    + partial score 4 in 1 simple subordinate character and 0 subordinate character hierarchies
      - simple subordinate character
          7: 1 transformation in 3 subcharacters (score 4)
  * root character hierarchy at 6: total score 8
    + 3 gains/losses in 1 subcharacter (score 4)
    + partial score 4 in 1 simple subordinate character and 0 subordinate character hierarchies
      - simple subordinate character
          7: 3 transformations in 1 subcharacter (score 4)
  * root character hierarchy at 6: total score 8
    + 3 gains/losses in 1 subcharacter (score 4)
    + partial score 4 in 1 simple subordinate character and 0 subordinate character hierarchies
      - simple subordinate character
          7: 2 transformations in 2 subcharacters (score 4)
```

The second line of this output are the scores of all characters (given as the product of prior weight and basic score when the prior weight differs from one). The interesting point about characters six and seven, part of the same

character hierarchy, is that only the total score of the character hierarchy is provided. This total score is given for the main absence/presence character of the hierarchy, character six in this case. The score for character seven is just a placeholder reference to character six. The reason is that, in general, character hierarchies have no unique distribution of their total score over their constituent characters. So in a brief summary as provided here, it only makes sense to give the score of the full hierarchy.

The rest of the output of statement 'characters score' gives a more detailed report of the score of the character hierarchy. It does so in terms of gains/losses, numbers of subcharacters, and transformations (in the upcoming beta I may change this to explained independent observed similarities). As it happens, all three possibilities have three gains/losses in the main absence/presence character and subscore 4 for dependent character seven. The difference is in the number and exact identity of the subcharacters or regions of applicability that are implied by the exact series of gains/losses in main character six, and consequences thereof for subordinate character seven.

These can be inferred from the output of statement 'characters optimize plot'. First consider character six. As can be expected from the above detailed report, the output of 'characters optimize plot' for character six consists of three different plots. In each of them, the inner node state reconstructions shown are not individual reconstructions but final statesets as commonly used for characters that have no inapplicables (they just happen to be singleton sets in this case):

```
characters optimize plot> plotting character optimizations on current tree
characters optimize plot> all leaves included
* character 6 on tree 1
  part of character hierarchy <6 7>
  nonaggregated final state sets for character 6
  ┌[0] B
  └[0]─┬[1] A
       └[0]─┬[1] C
            └[0]─┬[?] D
                 └[0]─┬[0] E
                      └[0]─┬[0] F
                           └[1]─┬[1] G
                                └[1] H
  ┌[0] B
  └[1]─┬[1] A
       └[1]─┬[1] C
            └[1]─┬[?] D
                 └[1]─┬[0] E
                      └[1]─┬[0] F
                           └[1]─┬[1] G
                                └[1] H
  ┌[0] B
  └[1]─┬[1] A
       └[1]─┬[1] C
            └[0]─┬[?] D
                 └[0]─┬[0] E
                      └[0]─┬[0] F
                           └[1]─┬[1] G
                                └[1] H
```

The second solution corresponds to the implied optimization in Brazeau et al.'s Fig. 3d, where three independent observed shared tail presences can be explained as due to inheritance and common descent. The first and the third solution arrive at the same grand total of explained observed similarities, but here some of those are shared observed tail absences.

The third solution, for example, can explain the shared tail presence in A and C, the shared tail presence in G and H, and the shared tail absence in E and F. No other shared observed similarities can be explained as due to inheritance and common descent on that optimization.

As an aside, note that the direct ancestor of D and (E F G H) is assigned state 0 in that third solution. That is correct by itself, but state 1 could be present there as well. That's one of a series of small issues that still need to be corrected before beta release.

The second 'characters optimize plot' statement gives the corresponding optimizations for character seven:

```
characters optimize plot> plotting character optimizations on current tree
characters optimize plot> all leaves included
* character 7 on tree 1
  part of character hierarchy <6 7>
  nonaggregated final state sets for character 7
```

```
┌[-] B
└[-]┬[0] A
   └[-]┬[1] C
      └[-]┬[?] D
         └[-]┬[-] E
            └[-]┬[-] F
               └[23]┬[2] G
                    └[3] H

┌[-] B
└[0123]┬[0] A
       └[0123]┬[1] C
              └[0123]┬[?] D
                     └[0123]┬[-] E
                            └[0123]┬[-] F
                                   └[0123]┬[2] G
                                          └[3] H

┌[-] B
└[01]┬[0] A
     └[01]┬[1] C
          └[-]┬[?] D
              └[-]┬[-] E
                  └[-]┬[-] F
                      └[23]┬[2] G
                           └[3] H
```

The second solution is the solution that is obtained with Brazeau et al's algorithm. The two other optimal solutions are new.

The last anagallis statement ('coppAbc 6.7') illustrates how anagallis statements can be arbitrarily abbreviated as long as the abbreviation can be uniquely resolved. In this case it is a 'character plot optimization' statement for characters six and seven and using options 'b' and 'A'. With that latter option, the different possible final statesets are aggregated to come to a more condensed representation. It comes at the cost that it is more difficult to reconstruct individual optimal reconstructions:

```
characters optimize plot> plotting character optimizations on current tree
characters optimize plot> all leaves included
* character 6 on tree 1
  part of character hierarchy <6 7>
  aggregated final state sets
  ┌[1] A
  └[01]┬[0] B
       └[01]┬[1] C
            └[01]┬[?] D
                 └[01]┬[0] E
                      └[01]┬[0] F
                           └[1]┬[1] G
                               └[1] H

* character 7 on tree 1
  part of character hierarchy <6 7>
  aggregated final state sets
  ┌[0] A
  └[0123-]┬[-] B
          └[0123-]┬[1] C
                  └[0123-]┬[?] D
                          └[0123-]┬[-] E
                                  └[0123-]┬[-] F
                                          └[0123]┬[2] G
                                                 └[3] H
```

Discussion

In this particular example some of the possible solutions that maximize explained shared observed similarity are not found by Brazeau et al's algorithm. The root cause seems to be a built-in constraint on the implicit evaluation of the implied absence/presence character(s). It is easy to see that this constraint may lead to an overestimation of the optimal minimal score with more complex character hierarchies. This, in turn, can lead to considering suboptimal trees as optimal during tree search, and to discarding trees that are optimal.

References

- Brazeau, M. D., Guillerme T., Smith, M.R. **2017**. Morphological phylogenetic analysis with inapplicable data. bioRxiv preprint first posted online Oct. 26, 2017. doi: http://dx.doi.org/10.1101/209775.

- De Laet, J. **2002**. Abstract. Parsimony algorithms for characters that are inapplicable in some terminals. XXIst Meeting of the Willi Hennig Society "21st Century cladistics". Helsinki, Finland, August 12-15, 2002. Abstract appeared in Cladistics 19: 148 (2003). [session *High power computing*]

- De Laet, J. **2003a**. Abstract. When one and one is not two: parsimony analysis of sequence data. XXIIth Meeting of the Willi Hennig Society. New York Botanical Garden, July 20-24i, 2003. Abstract appeared in Cladistics 20: 81 (2004).

- De Laet, J., and Wheeler, W. **2003b**. POY version 3.0.11. (Wheeler, Gladstein, and De Laet, May 6 2003). Command line documentation. 67 pp. [not peer reviewed]

- De Laet, J. **2005**. Parsimony and the problem of inapplicables in sequence data. Pp. 81-116 in Albert, V.A. (ed.) Parsimony, phylogeny and genomics. Oxford University Press, ISBN 0-19-856493-7.

- De Laet, J. **2012a**. Abstract. Theory and practice of parsimony analysis when some characters are inapplicable in some terminals. XXXIth Meeting of the Willi Hennig Society. Riverside, CA, June 23-27, 2012.

- De Laet, J. **2012b**. Presentation. Theory and practice of parsimony analysis when some characters are inapplicable in some terminals. XXXI Meeting of the Willi Hennig Society. Riverside, CA, June 2012. First posted online at Researchgate on July 20, 2017 DOI: 10.13140/RG.2.2.12159.51363.

- De Laet, J. **2013a**. Abstract. Anagallis: a program for minimization of homoplasy in characters that are inapplicable in some terminals. XXXIInd Meeting of the Willi Hennig Society. Rostock, Germany, August 3-7 2013.

- De Laet, J. **2013b**. Presentation. Anagallis, a program for minimization of homoplasy in characters that are inapplicable in some terminals. Presentation at the XXXII Meeting of the Willi Hennig Society. Rostock, Germany, August 2013. First posted online at Researchgate on July 20, 2017 DOI: 10.13140/RG.2.2.15514.95682

- De Laet, J. **2015**. Parsimony analysis of unaligned sequence data: maximization of homology and minimization of homoplasy, not minimization of operationally defined total cost or minimization of equally weighted transformations. Cladistics 31: 550-567.

- De Laet, J., Castroviejo-Fisher, S. **2016**. Parsimony analysis of unaligned sequence data: an exchange. Pdf version of an ongoing exchange in ResearchGate. Version 2. 20 pp. [not peer reviewed]

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

- Farris, J.S. **2008**. Parsimony and explanatory power. Cladistics 24, 825-847.

- Maddison, W. P. **1993**. Missing data versus missing characters in phylogenetic analysis. Syst. Biol. 42, 576-581.