WILEY-BLACKWELL

Cladistics 26 (2010) 453–455

Cladistics

10.1111/j.1096-0031.2010.00306.x

Letter to the Editor

A problem in POY tree searches (and its work-around) when some sequences are observed to be absent in some terminals

Accepted 8 January 2010

Sir,

The methods implemented in POY (Wheeler et al., 2003; Varón et al., 2010) include direct optimization (Wheeler, 1996) and fixed-states analysis (Wheeler, 1999), which are intended as heuristic techniques for calculating tree cost in the tree alignment problem (*sensu* Sankoff, 1975; Sankoff and Cedergren, 1983; see De Laet, 2005, pp. 97–99 for background and discussion). It develops that POY's implementation of those heuristics can give erroneous results in some cases. When the data include sequences or fragments that are absent in some terminals, POY may fail to count the indel events that are required to account for those absences. This can lead to misidentification of optimal trees and incorrectly resolved consensus trees. Here I describe the problem, provide a work-around, and discuss an example in which the issue has affected results with empirical data recently reported in this journal (by Agolin and D'Haese, 2009).

The basic problem is presented in the data set of Fig. 1a, with two short fragments for three terminals. The first fragment is present and identical in the three terminals, the second fragment is identical in the first two but absent in the third: this third terminal lacks anything comparable to the second fragment. To account for this data set on the single tree for three terminals, one indel event has to be postulated, on the branch leading to the third terminal. But when the absent sequence is represented as a zero-length string in the fasta input file for POY (Fig. 1b), POY reports a cost of zero, irrespective of the tree cost heuristic employed and the cost matrix applied. This is the case in both POY3 and POY4 (up to 4.1.2, the most recent version available).

The zero cost that POY reports would be correct if the second fragment in the third terminal were not an observed absence but missing data. The problem, then, is that POY sometimes treats absence as missing data. Information input as an observed absence of a sequence—as a zero-length string—is interpreted as missing data in the cost calculations. But the treatment of absences is not consistent. If the data are summarized with the *report(crossreferences)* command, POY lists the second fragment as absent for the third terminal. Similarly, when POY4[1] produces an implied alignment (Schwikowski and Vingron, 1997; cf. De Laet, 2005, pp. 98–99) for the data set of Fig. 1a, the third terminal's second fragment is presented as a gap, i.e. as an absent sequence, not as missing data.[2] This can be confirmed by using TNT (Goloboff et al., 2008) to evaluate the implied alignment on the single tree for three terminals. TNT returns a length of one unit indel, which is the correct length when absence is correctly treated as absence.

A work-around can be used to force POY to treat absences correctly. To each fragment that is absent in some of the terminals, add a zero-cost uninformative position in every terminal for which the fragment is non-missing. Figure 1c illustrates applying this method to

*Corresponding author:
E-mail address: Jan.De.Laet@skynet.be

[1]POY3 uses the character *X* throughout for the second fragment in the third terminal. *X*, a IUPAC code for any nucleotide, is interpreted as either any nucleotide or a gap in POY3—missing data, that is (this undocumented feature is easily verified with simple test data sets; alternatively, it is clear from the POY 3.0.11 source code file *utils.ml*, where all bits of the bitfield that stores character states are turned on for character state code *X*). By employing a non-standard definition of IUPAC code *X*, then, POY3 manages to interpret absence as missing data consistently.

[2]POY4's decision to output gap-only sequences for missing data in implied alignments (thereby turning missing sequences into absent sequences) seems to be motivated by the desire to keep the alignments readable by other programs (Varón and Cevasco, 2008, p. 7). In POY3, this general readability is maintained by using IUPAC code *X*.

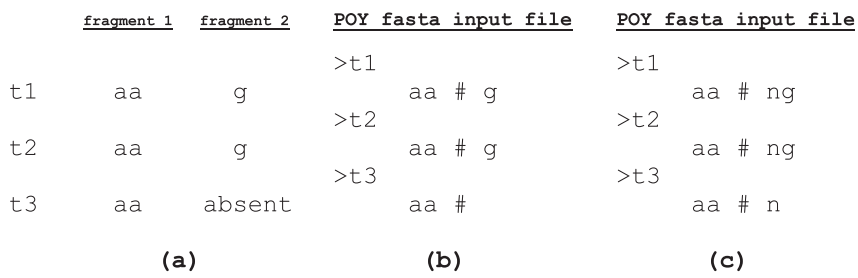

Fig. 1. A dataset and two possible renditions for its input in POY. (a) Distribution of two sequence fragments in three terminals, *t1–t3*; the second fragment is positively absent in the third terminal. On the single tree for three terminals, these data require a single indel of length one, on the branch that leads to the third terminal. (b) When the absent sequence is represented as a zero-length string, POY treats it as missing data and reports a cost of zero. (c) With the addition of an uninformative position to the fragment that is absent in some terminals, POY correctly treats the absence as absence and reports the cost of an indel of length one.

the data of Fig. 1a.[3] This turns each absence into the presence of just the dummy position, so that POY can no longer interpret the absence as missing data. A residue as added in Fig. 1a does not affect the (correctly calculated) cost of a tree alignment, so that any tree optimal for the augmented dataset is also optimal for the original data set. When applied to the augmented data set of Fig. 1c, both versions of POY report a length of one indel, just as they should have done for the original data set of Fig. 1a.

With more than three terminals, misinterpreting absences as missing data during tree search can lead to incorrect or incomplete identification of optimal trees. Some optimal trees are missed, for example, in the case illustrated by Fig. 2.

The problem that is brought to attention here has affected Agolin and D'Haese's (2009) analysis of the collembolan family Odontellidae. The sequences in their analysis are not nucleotide sequences but sequences of setae on a number of thoracic and abdominal segments. Agolin and D'Haese put forward the hypothesis that the setae in these rows are evolutionarily related through substitutions (of one *nature* of seta into another) and indel events (that change the *connections* among the setae), so that the problem lends itself to an analysis using direct optimization in POY4. The cost function that they use assigns a cost of one to any substitution and a cost of n to an indel of length n.[4] Under these conditions, Agolin and D'Haese (2009, p. 359) report 37 trees of cost 60 for the data set that describes the sequences of setae. Reanalysing their data, I obtained 50 trees of this length,[5] although still with the same strict consensus (their fig. 4). The lengths of the implied alignments for these trees, however, are not 60, but 62–65.[6]

The discrepancy between costs as reported by POY during tree search and evaluation (60) and the lengths of the implied alignments (62–65) is caused by setae row *m* of the fourth abdominal segment, a row in which no setae are present in five of the 26 terminals (Agolin and D'Haese, 2009, pp. 374–376, appendix 2). Because POY treats these observed absences as missing data during tree search, it does not count the indel events that lead to the absences, which is to say that it ignores part of the data during tree search. POY4 implied alignments, however, treat absences correctly, so that data are not ignored and this is why the cost increases. That some implied alignments have cost 62 whereas others have a higher cost means that some trees require more indel events to explain the observed absences than other trees.

Of course there is no guarantee that even the trees of cost 62 are optimal, let alone that all optimal trees have been found. After all, they are just a subset of trees that were obtained with some of the data discarded. The optimal cost can be found by applying the work-around

[3]There is a complementary work-around to make sure that POY4 implied alignments correctly reflect the interpretation of missing data where that is intended. Here, it is the implied alignment that needs editing, and all missing fragments must be changed from a long stretch of unit gap characters into an equally long stretch of whatever character indicates missing data in the program that will be used to read the implied alignment (e. g. *X* in POY3 or a question mark in TNT).

[4]Assigning an indel of length n the same cost as n substitutions amounts to the rather strong assumption that indel events only affect single positions at a time. De Laet (2005, pp. 112–114) provides a discussion of this well-known issue in the context of POY analyses.

[5]The different number of optimal trees found is interesting to note but has no relevance for the discussion of how POY treats observed absences. It may be explained by the version of POY and/or the search strategy that was used. Agolin and D'Haese, using POY4 beta 2635, included the script that they ran to obtain their 37 trees of cost 60. With the downloaded POY 4.1.2 binary for MacOS on a Mac Pro with 32 GB of RAM, this script made POY crash with a *malloc* error message. I therefore used a different script and obtained 50 trees of length 60. This script, the data set, and all other scripts and data sets used in this note are available upon request.

[6]Using TNT to evaluate each tree and its POY-generated implied alignment, three trees have a cost of 62, 27 of 63, 18 of 64, and two of 65. Alternatively, when using the implied alignments for tree searches in TNT, 45 result in length 62 and five in length 63.

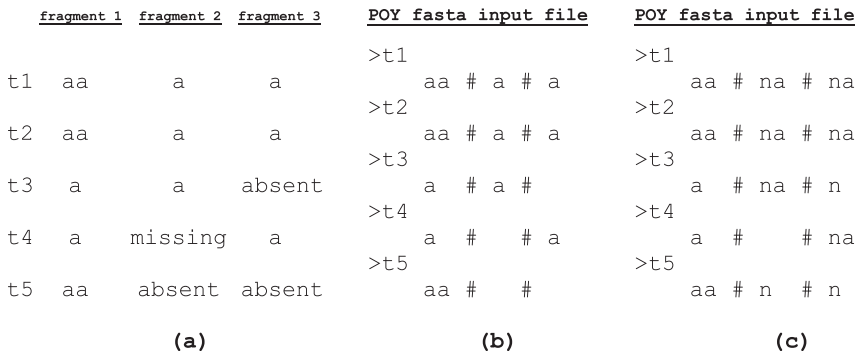

Fig. 2. A dataset and two possible renditions for its input in POY. (a) Distribution of three sequence fragments in five terminals, *t1–t5*; one terminal has a missing fragment ("*I don't know what should be put here, haven't properly done my sequencing yet*") and there are three cases of absence ("*I'm positive that this fragment is not present here*"). The two optimal solutions come at the cost of four indels of length one; one solution groups *t3* with *t4*, the other with *t5*. (b) When absent and missing fragments alike are coded as zero-length strings, POY treats both cases as missing data and only retrieves the solution that groups *t3* and *t4*, at the cost of one indel of length one. (c) With the addition of an uninformative position to fragments two and three in all terminals for which those fragments have been effectively scored, POY properly distinguishes missing data and a priori absence and finds both solutions at the correct cost.

to setae row *m* of the fourth abdominal segment. With the data so augmented, POY directly finds and reports the optimal trees using all data. In this case, the optimal cost does happen to be 62, and 15 trees of this cost are obtained. Compared to the strict consensus of Agolin and D'Haese, the strict consensus of the optimal trees has one additional node: *Brachystomella parvula* is recognized as sister to *Pseudostachia populosa* and *Pseudostachia xicoana*.

Acknowledgements

Thanks to Steve Farris for encouragement and constructive criticism and for putting mahayana at my disposal, the 4-core Mac with 32 GB of RAM that I used for the calculations in this note. The manuscript also benefitted from remarks by Pablo Goloboff.

References

Agolin, M., D'Haese, C.A., 2009. An application of dynamic homology to morphological characters: direct optimization of setae sequences and phylogeny of the family Odontellidae (Poduromorpha, Collembola). Cladistics 25, 353–385.

De Laet, J.E., 2005. Parsimony and the problem of inapplicables in sequence data. In: Albert, V.A. (Ed.), Parsimony, Phylogeny, and Genomics. Oxford University Press, Oxford, pp. 81–116.

Goloboff, P.A., Farris, J.S., Nixon, K.C., 2008. TNT, a free program for phylogenetic analysis. Cladistics 24, 774–786.

Sankoff, D., 1975. Minimal mutation trees of sequences. SIAM J. Appl. Math. 28, 35–42.

Sankoff, D., Cedergren, R.J., 1983. Simultaneous comparison of three or more sequences related by a tree. In: Sankoff, D., Kruskal, J.B. (Eds.), Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison. Addison-Wesley, Reading, MA, pp. 253–263.

Schwikowski, B., Vingron, M., 1997. The deferred path heuristic for the generalized tree alignment problem. J. Comput. Biol. 4, 415–431.

Varón, A., Cevasco, M. 2008. POY 4.0 Tutorials: General Commands. Available at http://research.amnh.org/~avaron/poy/tutorials.

Varón, A., Vinh, L.S., Wheeler, W.C., 2010. POY version 4: phylogenetic analysis using dynamic homologies. Cladistics 26, 72–85.

Wheeler, W., 1996. Optimization alignment: the end of multiple sequence alignment in phylogenetics? Cladistics 12, 1–9.

Wheeler, W., 1999. Fixed character states and the optimization of molecular sequence data. Cladistics 15, 379–385.

Wheeler, W.C., Gladstein, D., De Laet, J. 2003 POY, Phylogeny Reconstruction via Optimization of DNA and Other Data, Version 3.0.11. Available at ftp://ftp.amnh.org/pub/molecular/poy.

Jan De Laet*
Gothenburg Botanical Garden,
Carl Skottsbergs Gata 22 A,
SE – 413 19 Gothenburg,
Sweden