Zootaxa 3641 (5): 587–594
www.mapress.com/zootaxa/

Article

ISSN 1175-5326 (print edition)
ZOOTAXA
ISSN 1175-5334 (online edition)

http://dx.doi.org/10.11646/zootaxa.3641.5.7
http://zoobank.org/urn:lsid:zoobank.org:pub:270BD980-0983-4AE9-AB53-755CE4A68611

Implementation as theory, hierarchy as transformation, homology as synapomorphy

MALTE C. EBACH[1], DAVID M. WILLIAMS[2] & TEGAN A. VANDERLAAN[1]

[1]*School of Biological, Earth & Environmental Sciences, The University of New South Wales, Kensington, NSW, Australia.*
E-mail: mcebach@unsw.edu.au, t.vanderlaan@unsw.edu.au
[2]*Department of Life Sciences, the Natural History Museum, Cromwell Road, London SW7 5BD, United Kingdom.*
E-mail: d.m.williams@nhm.ac.uk

Abstract

Presently cladistics is based on, and justified by, a computer implementation (Wagner Parsimony) rather than sound methodological or theoretical foundations. In this paper, the recent goals of cladistics that are derived from computer implementations and 'tree-thinking' are discussed. The discussion raises a critical point, namely whether one can dispense with these recent goals and adopt a wholly systematic approach, one herein termed systematic thinking.

Key words: Cladistics, computer programs, homology, synapomoprhy, systematic thinking, tree thinking

Introduction

Cladistics developed from the works of Willi Hennig (1950, 1966) and Lars Brundin (1966) and has, over the last 30 years, created several conflicting hypotheses, but with one overarching methodology, namely Wagner Parsimony (see Williams & Ebach 2008; Schuh & Brower 2010).

Divorced from such methodology, cladistics is founded on a simple premise: two or more taxa are more closely related to each other than they are to any other taxon. Where this premise conflicts is when the term relationship (i.e., homology) is defined. This is clearly illustrated in the definition of homology given by Brower and de Pinna (2012):

> "*Homology is the relationship among parts of organisms that provides evidence for common ancestry*" (Brower & de Pinna, 2012, p. 537, their italics)

Brower and de Pinna offer four properties:

> "This definition expresses the real world praxis of phylogenetic inference and applies to both layers of homology assessment, primary and secondary. Also, it places evidence where it belongs: before models. It does not specify that a hypothesized relationship is corroborated through parsimonious accounting of character-state transformations, although that is the method that we advocate. Nor does it assume a priori an evolutionary causal mechanism, although it implies that evolution is the best explanation for the observed pattern" (Brower & de Pinna 2012, p. 537)

While none of this seems controversial, their final 'property' offers scope for further discussion:

> "Those familiar with phylogenetics are welcome to abbreviate our definition by replacing 'synapomorphy' for all that follows the verb 'is'." (Brower & de Pinna, 2012, p. 537)

That is:

Homology = Synapomorphy.

Accepted by M.R. De Carvalho: 11 Mar. 2013; published: 29 Apr. 2013

Increasingly, cladistics is viewed in terms of algorithmic implementation via one or another computer program (e.g., TNT Goloboff, 1999), which aim to convert any binary data matrix into a branching diagram (or diagrams) purportedly summarising the original data economically. In doing so a crucial connection has been lost with the overall methodological goals, namely to find homologous relationships. That connection can be rediscovered if solutions (branching diagrams) were arrived at by hand (non-algorithmic implementation), with no other tool than pencil and paper. While this may sound simple enough, the underlying cladistic methodology, originally founded on non-algorithmic implementation, is lost when moving from observed parts to binary data matrices and computer programs.

Binary data matrices that represent data have no foundation in cladistic methodology and bear little resemblance to what is done by hand, that is, the original cladistic methodology prior to algorithmic implementation. But these are now treated as a standard and have replaced a whole methodology with a computer algorithm along with attempts to justify this based on history and theory (see Nixon & Carpenter 2012). Here a controversy can be appreciated: confusing methodology with algorithmic implementation versus confusing algorithmic implementations with homology. The result is a numerical cladistic worldview in which the computer algorithm *is* the methodology, which conversely underpins a theory. We challenge this worldview by presenting a recent debate which has changed the way homologies are interpreted in cladograms (Brower & de Pinna 2012; Nixon & Carpenter 2012).

The overall methodological goals in current cladistics are based solely on how the systematist handles and analyses data using a computer program (e.g., compiling data matrices, selecting out-groups and generating branch lengths), rather than on a sound methodological or theoretical foundation (i.e., understanding and discovering homology and monophyly). Moreover, the recent goals of cladistics are derived from 1) computer programs or implementations and; 2) a posteriori interpretations of branching diagrams. The point we wish to raise is whether we can dispense with computational procedures and pursue a methodological and theoretical goal.

Below two separate points are raised in order to clarify the proposed position as well as providing definitions to assist with outlining that clarity.

Position 1: The fundamental goals of cladistics have been defined by computer programs

Definition of terms—Fundamental goals refers to the theoretical underpinnings of homology and monophyly. Such theory may be implemented using a methodology, the outline of a theoretical process, for example, Hennig's grouping by synapomorphy for systematics (Hennig 1966) or Nelson's Assumption 2 for biogeographic analysis (Nelson & Platnick 1981). By algorithmic implementation we refer to the conversion of a method into an algorithm (or a set of algorithms) to process data by a computer under the criteria outlined in the methodology. For example, Hennig's grouping by synapomorphy may be implemented by Wagner parsimony (Farris 1970), Nelson's Assumption 2 may be implemented by Component Analysis (Page 1993).

Implementation as Theory

The concept of homology concerns sameness rather than similarity. (Wilkins & Ebach in press). rather than similarity. By sameness we mean different manifestations of the same thing: a bat's wing and a whale's fin are both manifestations of fore-arms. Similarities are simply likenesses or measurements (that is they are phenetic). Any two things can be considered similar in some way, while manifestations of the same thing are homologous. Although sameness between homologs can be observed, it is tested using a methodology.

While this might not be surprising, it does raise some points. For example, do computer programs actually replicate methods correctly, are they finding homologies? Do binary data matrices correctly represent homologs? We believe they do not. Rather, computer programs process an unstructured data matrix that contains a cluster of entries (e.g., binary data). Data that match are the similarities, which the program sorts into sets (i.e., structured data). As such, current computer programs are only able to implement various phenetic approaches (i.e., similarity), rather than the intended methodology conceived through non-algorithmic implementation (i.e., sameness or affinity). However, this does not mean that the methodology computer programs represent are themselves necessarily phenetic. Rather, the degrees of co-ordinated similarity are taken as an approximation of actual hypotheses of relationship. While it is true for all computer programs that attempt to find similarities, it is not controversial.

Computer programs are tools and do not recover real phylogenetic trees, actual schemes of ancestry and descent. Rather, they recover branching diagrams that are interpreted by the systematist as phylogenetic trees: the branching diagrams = a phylogenetic tree and; a node = hypothetical ancestor, and so on (Fig. 1a). Cladistics is not phenetics, however "cladistic" computer programs implement a phenetic approach. This claim is not controversial either, so why has it become a *cause célèbre* within the cladistics community?

We suspect that proponents of computer algorithms have confused the implementation, namely the software, as cladistics or its equivalent. If so, then any criticism of a popular computer program (e.g., TNT) or implementation (e.g. Wagner Parsimony) is instantly a criticism of cladistics *per se*. If this confusion between computer programs and cladistics were extrapolated further, then cladists would create a methodology and theory based on what computer programs do rather than what the original cladistic theory states. This has resulted in two very different approaches, namely pattern and numerical cladistics. The latter approach derives its aims and goals from an algorithmic implementation or computer program, while the former derives it from theory from earlier systematic morphological theory (see Williams & Ebach 2008).

As a result, numerical cladistics has become problematic in two ways. Firstly, it has created a version of homology we feel is derived from a how a computer program functions. For instance, de Pinna (1991) proposed primary and secondary homology from a reading of Patterson's homology tests (Patterson, 1982). Yet Patterson's homology tests are a means with which to sort types of similarity and are not tests for homology. Furthermore, de Pinna (1991) is an extrapolation of this same set of tests but seen through the lens of the binary matrix and Wagner parsimony: primary homology = similarly of data; the resulting branching diagram as an example of secondary homology = congruence of data.

Secondly, many observations made on organisms are now tailored to fit data entry in the typical binary matrix format. For instance, we no longer "study homologs", instead we "code characters". (Interestingly, in molecular systematics characters are neither seen nor coded; they just are.) Given that greater emphasis is now placed on implementation, the term "zero" has entered the cladistic worldview to generally mean "primitive", alongside other purely functional terms, such as "coding", "matrix", "absence", "presence", "branch length", all of which have some meaning beyond their original intent. Moreover, we now confuse the plethora of terms ("coding", "bootstrap", "parsimony" and so on) to be cladistic, ignoring the fact that these are terms and techniques derived from elsewhere. This practice has degraded cladistics, once a developing theory and method of classification, to be synonymous with a suite of computer programs. Therefore, recent debate revolves around computer programs rather than method and theory. It almost seems as if practitioners of cladistics, such as Nixon and Carpenter (2012), are creating theory and history to justify their choice of computer program. For example, consider de Pinna's primary and secondary homology (de Pinna 1991). If homology is simply the relationship between homologs, then why is a functional two-step approach required? "Primary" is designed to delimit data in the form expected by computer programs (the ones and zeros listed that adorn matrices), "secondary" to interpret and overturn the primary data (secondary homology being merely the result of the implementation of one or another model).

Position 2. We believe that the recent goals of cladistics are derived from interpretations of branching diagrams

Definition of terms—Monophyly is defined as a relationship, namely that two taxa are more closely related to each other than they are to a third. Non-monophyly denotes a group whose component taxa share closer relationships with various other taxonomic groups than they do other. Note that monophyletic taxa can be grouped, but are in themselves not necessarily groups. Moreover, inclusion of a most recent common ancestor is not relevant to this definition of monophyly. Common ancestry is an interpretation derived from monophyletic groups after their discovery, not before.

Tree thinking is another term we wish to define. O'Hara (1997) defines the term:

> "Tree thinking, in contrast to group thinking, considers species in a phylogenetic context, not as independent replicates but as parts of a single phylogenetic tree. If we seek to understand common causes acting in evolution then the replicates we need to examine are not species; but the evolutionary events that are of interest in a particular study, and this can only be done by plotting those events on a tree" (O'Hara, 1997: 325).

O'Hara's definition may be simplified as "interpreting branching diagrams as evolutionary trees (phylogenies) in which nodes represent ancestors (plesiomorphies) or transformations".

Hierarchies as Transformations

Cladograms and phylogenetic trees are extrapolated from the branching diagrams produced by computer programs (i.e., tree-thinking) (Fig. 1c, d). The structure of the branching diagram and the position of binary data at the nodes are interpreted to mean various things such as ancestral relationships and transformations. Computer programs do not compute nor do they interpret transformations, they simply calculate similarities within a hierarchical cluster. The notion of transformation, however, comes from elsewhere (i.e., ontogeny). The way cladists apply it to branching diagrams is to follow a simple rule: if a character-state "changes" within a hierarchy (which it must do in order for a hierarchy to be there in the first place) it is a transformation. For example, the branching diagram A{B,C} contains a single character C1 (Fig 1c). The relationship between the character-states is 0{1,1}, namely a hierarchy of character-states. However, this hierarchy is often interpreted via tree-thinking to be a linear transformation, namely 0 → 1. While the interpretation is harmless enough, it is often mistaken to be evidence within tree-thinking: without a hierarchy there is no branching diagram. Understanding this to be evidence for transformation is erroneous as transformations are both linear and interpreted solely from branching diagrams a posteriori to any analysis. In this sense, transformations are clearly unobservable and highly speculative.

The above claim does not mean transformations themselves do not exist. In fact, numerous arguments against Wagner Parsimony have been dismissed as non-evolutionary because cladists confuse Wagner Parsimony as cladistics rather than one of its many implementations. Rather, cladists misinterpret hierarchies in branching diagrams as evidence of linear transformations. The multiple misdirections made by several numerical cladists perpetuate the misunderstanding of what branching diagrams are and how they function. If one allows a computer program to group a single character-state more than once, for instance, one ends up with expanded hierarchies. For example, the branching diagram A{B{C,D}} may contain an expanded character-state hierarchy 0{1{0,0}} (Fig 1c). If this is allowed to occur, then data can be artificially stretched further. If this were not to occur, then the following trichotomy would be recovered: B{A,C,D} based on the homolog relationship 0{1,0}. In neither case is the branching diagram created by a transformation. Any transformation is interpreted a posteriori to analysis (Fig 1d). The computer program does not search for transformations, rather it builds hierarchies of character-states based on certain parameters (i.e., such as allowing for character-state 0 to be grouped more than once per character). Viewing a computer implementation in this way, one would be wary of such an extended grouping as it creates resolution where there may be none.

Methodologically, however, transformations can be represented as hypotheses. If, for instance, a secondary loss may have occurred, then the relevant character-state can be coded as such. In this case another character-state (i.e., to represent a secondary loss) may be added in and tested in the usual manner.

A computer program, however, finds a pattern based on similarity. It does not, and can never find, an evolutionary process such as transformation.

Homology as a synap[symplesio]morphy redux

Tree-thinking is mostly seen in how one interprets homology and homologs (the evidence) within branching diagrams. However, since cladistics became synonymous with computer programs, discussions of homology have mostly focused on the way one might code characters in a binary data matrix for processing with a particular program. These might be reduced to two possibly complementary views, both claiming, with some justification, illumination from the works of Willi Hennig (1950, 1966). The first, or implementational view, understands homology to be composed of two parts: synapomorphy and symplesiomorphy (Nixon & Carpenter, 2012). In this they follow Hennig. The alternative viewpoint, recently outlined by Brower and de Pinna (2012) suggests that homology is equivalent to synapomorphy, as symplesiomorphy is simply synapomorphy (homology) at another, higher level. The origin of that viewpoint (homology = synapomorphy) is often attributed to Patterson (1982), rather than Hennig, but disputed evidence suggests Hennig may have implied such a view.

One logical extension of Patterson's view is that symplesiomorphy and synapomorphy are no longer necessary terms: only homology has any evidential value. How does the implemetational view differ from that of Patterson (1982)? One might represent the implementational view so:

Homology = Symplesiomorphy (0) + Synapomorphy (1)

For Patterson (1982) and Brower and de Pinna (2012), however, one might represent their view so:

Homology = Synapomorphy (1)

or;

Homology = [Symplesiomorphy (0) → Synapomorphy (1)]

The numbers (0 and 1) following the words symplesiomorphy and synapomorphy above represent the traditional codes used in matrices for computer programs to indicate these different conditions: 0 representing symplesiomorphy, 1 representing synapomorphy. Further examination of the two viewpoints yields the following comparison:

Implementational: Homology = Symplesiomorphy (0) + Synapomorphy (1)
Patterson (1982): Homology = Symplesiomorphy (0)→ Synapomorphy (1)

From the point of view of determining relationships and the evidential value of synapomorphy and symplesiomorphy, they appear identical as both would yield the following relationship upon analysis: 00{1,1}. Why, then, do the authors consider them different and conflicting approaches? As far as one can tell, the primary issue centres on how computer programs manipulate the coded data (the 0s and 1s). Of course, in that respect the two views are not identical when examined more closely:

Implementational: Homology = Symplesiomorphy? (0) + Synapomorphy? (1)

The added question marks indicate a measure of uncertainty, as the characters are not assigned specific meaning prior to analysis. This allows the 0 or 1 to be(come) either symplesiomorphy or synapomorphy depending on how the program ‘fits’ the data to an optimal tree.

For the alternative the matter is a little trickier:

Homology = Synapomorphy (1)

Reversals are more problematic, yet one might allow for that outcome by disregarding the potential evidential value of the 0.

Homology = Synapomorphy?

To allow for the outcome desired by the implementational view one need only promote Wagner Parsimony as the justified “cladistic” program; for Patterson, a strict interpretation of his view on homology testing might instead invoke a compatibility approach. Patterson certainly thought so (Patterson 1982). In fact, others have recognised a similar generalisation with respect to characters and parsimony. For example, some years ago, Sankoff noted:

> “…maximizing parsimony is equivalent to minimizing the total number of extra steps over all characters. Maximizing compatibility is equivalent to minimizing the number of characters requiring at least one step” (Sankoff in Le Quesne 1975: 426).

The difference in both approaches is related to particular kinds of algorithms, which most likely means that they are dependent on individual implementations rather than on differences in methodology. In this sense there is a methodological (derived from theoretical foundations) and an implementational (derived from computer programs) definition of symplesiomorphy and synapomorphy.

Methodologically a synapomorphy is a group of homologs, which may be interpreted to be derived under some methods prior to analysis. Implementationally, synapomorphies are simply points of data located within a hierarchy interpreted as transformations post-computer analysis (Fig. 1e). While the distinction may be acceptable in morphology, it is not in molecular systematics, where no hypothesis is made prior to analysis. Implementation to one side, the distinctions between current methods in cladistics are equally divergent in how they distinguish between relationship, that is homology. Take for instance Hennig’s Phylogenetic Systematics:

Synapomorphy = group of homologs + relationship.
Relationship = non-hierarchical transformations (i.e., transformative relationships) within any given branching diagram (a.k.a. Special Similarity).

Here synapomorphy is a group of homologs that defines a relationship between taxa. Homology is considered to be a transformation from a primitive to derived homolog (transformative relationship) in any branching diagram. Numerical cladists have a different interpretation to Hennig due to their adherence to a specific computer program:

Synapomorphy = group of homologs + relationship.
Relationship = transformative relationship derived from an extended hierarchy.

The difference between numerical cladists and Hennig is their interpretation of relationship, that is, homology. Since the computer algorithms can only implement a phenetic approach, they rely solely on some form of similarity (homology = similarity). Moreover, in order to facilitate a reversal, an extra step is required to accommodate an extended hierarchy (e.g., 0{1{0,0}}). While this introduces an extra (or non-parsimonious) step, it ensures that a reversal can be interpreted from the branching diagram. In other words extended (non-parsimonious) hierarchies are reversals. In numerical cladistics homology is equivalent to congruent similarities between groups of primitive homologs (i.e., symplesimorphies) and derived homologs (i.e., synapomorphies).
Patterson suggested an almost purely phenetic approach:

Synapomorphy = group of homologs + relationship.
Relationship = similarity (or Special Phenetics).

Special Phenetics here means a two-item statement based on Patterson’s similarity test. Like Hennig, the synapomorphy consists of a group of homologs, however, without a notion of primitive (symplesiomorphy). Homology is equivalent to congruent similarities between derived homologs (i.e., synapomorphies).

Nelson (2004), however, has completely revised the search for homology:

Synapomorphy = a group of homologs within a branching diagram.
Relationship = hierarchical relationship of homologs only.

In Nelson’s method, the relationship (i.e., homology) is a hierarchy of homologs that are congruent with other such branching diagrams. In this sense, synapomorphies consist only of derived homologs. Nelson’s method assumes a hypothetical relationship prior to analysis (whereas all other methods do not), indicating that it is an improvement on Hennig’s method rather than seeking to implement ‘phylogenetic systematics’.

Systematic Thinking

The fundamental difference between all of these methods is the interpretation of the meaning of the nodes and the steps between taxa. Any step within a branching cladogram (with possibly the exception of the terminals) attempts to extrapolate knowledge beyond evidence (i.e., assumptions). An evolutionary biologist might interpret that one step as an ancestor; a Hennigian as a plesiomorphy or apomorphy; a phylogenetist as an allopatric, sympatric or parapatric event. The list is seemingly endless. The question proposed herein is: what evidence does one have to accept (or reject) these interpretations? The data contained within, the evidence, merely gives us a group of character-states. This one step interpretation is a result of tree-thinking. While branching diagrams might invite an obvious conclusion of “monophyly”, many of us wish to see beyond the systematic quest for classification. But what if this is the only evidence one has, namely evidence for monophyly and nothing else?

Cladistics is a robust but limited tool. Tree thinking extends branching diagrams beyond the evidence, creating mere speculation. This claim does not deny any evolutionary process. Rather, it denies the systematist the ability to discover them using what is effectively an array of molecular and morphological parts and a computer program to order them.

The demand for more elaborate evolutionary scenarios is based on a limited form of evidence, rather than a limited number of homologs, and is a result of tree-thinking. Instead 'systematic thinking' is proposed, in which there is a search for monophyly (see definition above), non-monophyly being discarded as irrelevant. Systematic thinking tests for natural classifications, which help us towards understanding taxonomic groups.

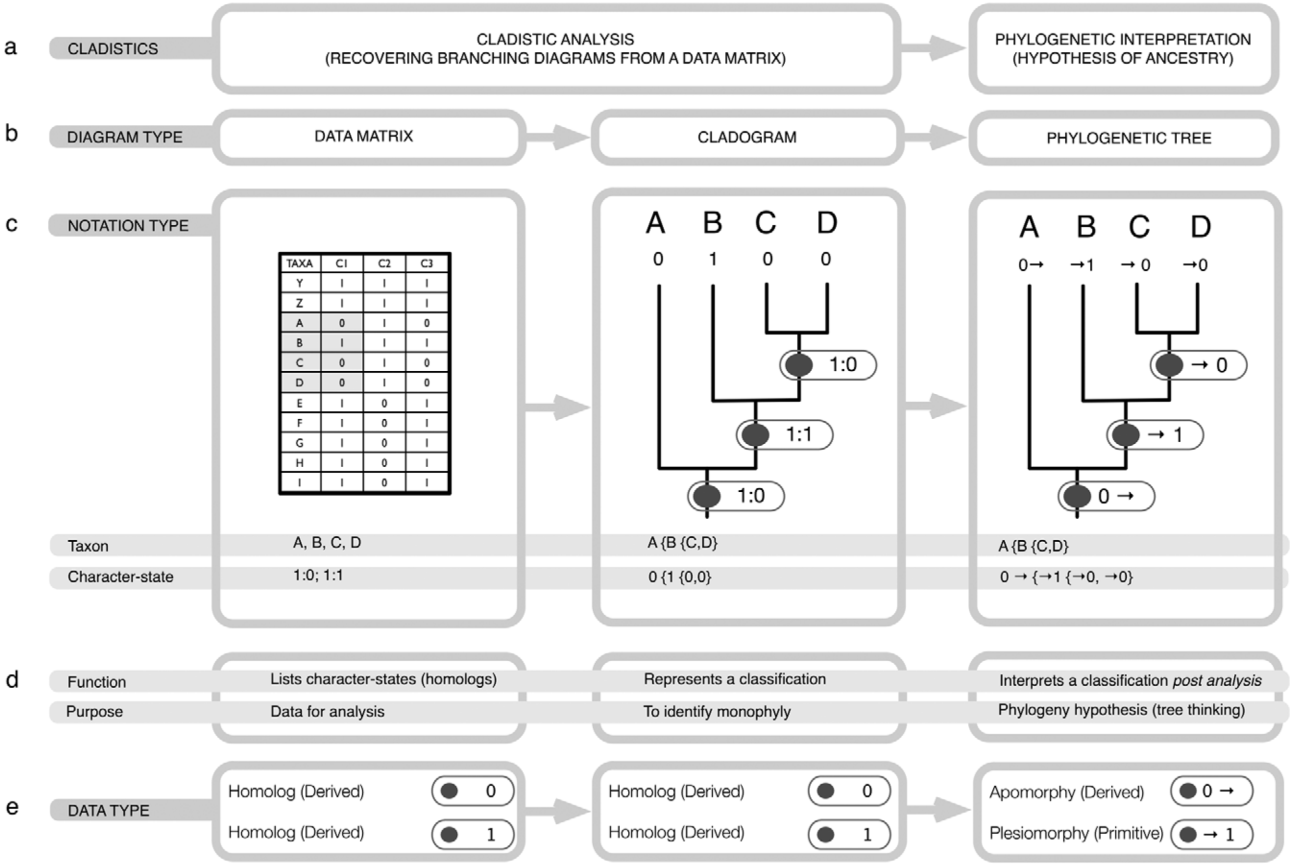

FIGURE 1. The anatomy of a cladistic analysis. a. A cladistic analysis recovers branching diagrams (cladograms) from a data matrix (e.g., binary or parenthesis matrix). The characater-state relationships (homologs) may be interpreted phylogenetically as transformations; b.The data matrix is analysed by a computer program that produces a cladogram. The phylogenetic tree is created through human interpretation only; c.#A data matrix contains ordered data, which is converted to a branching diagram (cladogram) using a computer program. The cladogram depicted here is based on character 1 (namely, a character tree). The character-states are treated as synapomorphies within A{B{C,D}, where C and D share character-state 0 or, the states can be shown as a relationship, namely 0{1,1}. In the phylogenetic tree, the character-states are shown as grouped plesiomorphies and apomorphies. The transformation is inferred by the person viewing the tree; d. The function of the data matrix is to show which character-states are ascribed to taxa. The cladogram represents a classification in order to identify monophyly, while a phylogenetic tree interprets a classification through transformation; e. the data matrix and cladogram represent homologs. In the phylogenetic tree, homologs are interpreted to be derived or reversed (apomorphic) or plesiomorphic (primitive).

Conclusion: Doing Away with Tree Thinking within Natural Classification

Systematic thinking approaches the question of homology from its original source, by asking what the relationships are between homologs and whether they are indeed homologous. From this perspective a systematic emphasis is retained, rather than interpreting evolutionary processes post-computer implementation (i.e., tree-thinking). For instance, the goal of systematics is to find patterns in branching diagrams. These patterns reveal themselves as homology and monophyly (see definition above). Any notion of ancestor-descendant relationship (at the character-state level) is created by the cladist post-analysis through interpreting hierarchies as transformations. Since computer programs are unable to comprehend the notion of evolution (they merely group by similarities), they are therefore unable to test for transformations. This means any assumed "hypothesis" prior to analysis is moot. For instance, the process of computing unstructured data can be represented as so,

Unstructured data (binary matrix) → Similarity Analysis (clustering) → Structured data (branching diagrams) → Interpretation (transformations).

Computing unstructured data, however, is often confused with hypothesis testing, namely:

Homologs (binary matrix) → Transformations (similarity analysis) → Phylogenetic trees (branching diagrams).

No hypothesis of transformation is tested. Rather a pattern (homology, monophyly) can be found within the structured data. Any a posteriori phylogenetic "inferences" are based on no evidence at all. They are merely well meaning interpretations or guesses.

Systematics, as the study of natural classifications, is independent of evolutionary narratives; phylogenetics is wholly dependent on systematics. With this in mind, the goal of any cladistic analysis is biological classification. Moreover, phylogenetic interpretations are dependent on such biological classification

Tree-thinking and confusing cladistics as a computer implementation has caused the 'arrested development' in the progress of understanding biological classifications (see Nelson, 2004). If tree thinking is dispensed with and systematic thinking embraced, both taxonomy and stable classifications are advanced, namely those items on which evolutionary biologists base their hypotheses.

Acknowledgements

Thanks to Anthony Gill for reading through earlier drafts. We also thank Andy Brower and Mario de Pinna for their support although their views on cladistics run contrary to ours. TAV wishes to thank the University of New South Wales for providing an Australian Postgraduate Award.

References

Brower, A.V.Z. & de Pinna, M.C.C. (2012) Homology and errors. *Cladistics*, 28, 539–544. http://dx.doi.org/10.1111/j.1096-0031.2012.00398.x

de Pinna, M.C.C. (1991) Concepts and tests of homology in the cladistic paradigm. *Cladistics*, 7, 367–394. http://dx.doi.org/10.1111/j.1096-0031.1991.tb00045.x

Goloboff, P.A. (1999) Analyzing large data sets in reasonable times: solutions for composite optima. *Cladistics*, 15, 415–428. http://dx.doi.org/10.1111/j.1096-0031.1999.tb00278.x

Hennig, W. (1950) *Grundzüge einer Theorie der phylogenetischen Systematik*. Deutsche Zentralverlag, Berlin, 370pp.

Hennig, W. (1966) *Phylogenetic Systematics*. University of Illinois Press, Urbana, 263 pp.

Le Quesne, W.J. (1975) The uniquely evolved character concept and its cladistic application. *Systemtatic Zoology*, 23, 513–517. http://dx.doi.org/10.2307/2412469

Nelson, G. (2004) Cladistics - its arrested development. *In*: Williams, D.M. & Forey, P.L. (Eds.) *Milestones in Systematics. Systematics Association Special. Vol. 67*. CRC Press, Boca Raton, Florida, pp. 127–147. http://dx.doi.org/10.1201/9780203643037.ch6

Nixon, K.C. & Carpenter, J.M. (2012) On homology. *Cladistic*s, 28, 160–169. http://dx.doi.org/10.1111/j.1096-0031.2012.00432.x

O'Hara, R.J. (1997) Population thinking and tree thinking in systematics. *Zoologica Scripta*, 26, 323–329. http://dx.doi.org/10.1111/j.1463-6409.1997.tb00422.x

Page, R.D.M. (1990) Component analysis: A valiant failure? *Cladistics*, 6, 119–136. http://dx.doi.org/10.1111/j.1096-0031.1990.tb00532.x

Patterson, C. (1982) Morphological characters and homology. *In*: Joysey, K.A. & Friday, A.E. (Eds.), *Problems of Phylogenetic Reconstruction*. Academic Press, London, pp. 21–74.

Schuh, R.T. & Brower, A.V.Z. (2010) *Biological Systematics: Principles and Applications, 2nd Edition.* Cornell University Press, Ithaca, New York, pp. 311.

Wilkins, J.S. & Ebach, M.C. (in press) The Nature *of Classification: Relationships and Kinds in the* Natural Sciences. Palgrave Macmillan, Basingstoke.

Williams, D.M. & Ebach, M.C. (2008) *Foundations of Systematics and Biogeography.* Springer, New York, pp. 309. http://dx.doi.org/10.1007/978-0-387-72730-1

Williams, D.M., Ebach, M.C. & Wheeler, Q.D. (2010) Beyond belief: the steady resurrection of phenetics. *In*: Williams, D.M. & Knapp, S. (Eds.), *Beyond Cladistics: The Branching of a Paradigm.* University of California Press, Berkeley, pp. 169–197.