Multi-purpose descriptive databases

11 September 2016

M.J. Dallwitz

Preface

This discussion was originally posted on the TDWG-SDD discussion list in July 2000 under the subject 'SDD standard - purpose'. See

lists.tdwg.org/pipermail/tdwg-content/2000-July/001390.html (Kevin Thiele, 20/7/2000)

lists.tdwg.org/pipermail/tdwg-content/2000-July/001370.html (Mike Dallwitz, 27/7/2000. Also posted on DELTA-L.)

Contents

Posting by Kevin Thiele on the TDWG-SDD discussion list

Initially, I think we should aim at descriptions and interactive ID. The idea of massaging one data file into two or more different products (e.g. natural-language and keys) is very attractive, but surprisingly problematical, since the structure of data needed for the two purposes is often subtly different. Of course, doing just this is the basis of the DELTA system, but we may need to do it in a more sophisticated way. ... The problems inherent in the multiple-product model become even more alarming when you try to maintain one data file for both description/identification and phylogenetic analysis. My personal view is that we should leave cladistics out of the scope at least for the time being.

Response by Mike Dallwitz

Introduction

It's not particularly difficult to accommodate description, identification, and phenetic or phylogenetic analysis in a single database.

By 'not particularly difficult' I don't mean that it's easy, particularly without the help of an experienced teacher. It's of comparable difficulty to many other aspects of professional work, for which we usually prepare by undertaking a degree course. Nor should we expect that advances in software will ever make it easy, in the sense that it could be done well without aptitude, training, thought, and experience. (In fact, software advances often make tasks *more* difficult, as greater capabilities lead to higher expectations.)

In addition to the obvious benefits of making as much use as possible of laboriously acquired data, there can be valuable synergies between the different kinds of application. For example, even if the data are primarily for phylogenetic analysis, using them for description and identification can help detect errors. It is not unheard of for published work to contain gross errors (such as frame shifts caused by the

accidental deletion of matrix elements) which could easily have been detected in this way. Also, the information-retrieval functions of Intkey can help in exploring patterns and relationships in the data.

Special-purpose characters

Within a given project, it's possible to define a 'universal' set of characters which are suitable for all applications. To these can be added characters designed for particular purposes, which are to be omitted for other purposes. For example, classification characters (e.g. the family to which a taxon belongs) and geographical distribution characters (what countries, states, etc. a taxon occurs in) are useful in description and identification, but would not normally be used in classification (for want of a better word, I will use this as an abbreviation for 'phenetic and phylogenetic analysis').

Sometimes it will be necessary to define alternative characters to represent similar concepts for different purposes. Obviously, efforts should be made to keep such alternative characters to a minimum. Software can help by combining character states, converting numeric characters to multistate, and checking the scoring of characters against relationships defined between them (not done by any current software as far as I know, except for the special case of character dependencies, which are checked by the CSIRO DELTA programs).

While some 'identification' characters are unsuitable for classification, the converse is not true. To claim that a 'classification' character is not suitable for identification is tantamount to an admission that the author's scoring of the character is not reproducible by others (or that the data have been 'sanitized' by omitting some character values). Of course, I am referring to interactive identification, using a program with a 'best characters' calculation fast enough to be used routinely, and supporting character weights.

Flagging state values for different purposes

With a given set of characters, it may be necessary to record attributes (i.e. the cells of the taxa × characters 'matrix') so that different state values can be used for different purposes. For example, in Lucid it is possible to flag state values as 'present by misinterpretation'. Values so flagged would normally be used for identification but not for description.

Our proposed new features for the DELTA System contain more general methods of flagging values for use in any number of user-defined applications. For example, consider the coding

16,2/1<@only keys> 17,7<@only keys>-8.5-9<@for classification>-10-12<@only keys>
18,2<@for classification>/3 20,1/2<@not Australia>

For the application 'keys', this would be interpreted as

16,2/1 17,7-12 18,2/3 20,1/2

and for the application 'classification Australia' as

16,2 17,9 18,2 20,1

Prototypes for using two such method of flagging state values are available in the DELTA System.

dismis.bat. Removes from an 'items' file state values that are 'present by misinterpretation'.

notckeys.bat. Removes from an 'items' file values not to be used for 'conventional' keys (as opposed to interactive keys).

Alternative character wordings

It is often necessary to use alternative *wordings* of characters for different purposes. (This is different from the alternative character *concepts* discussed above.) This arises: (1) because of the different contexts in which the words are used; (2) because of the different audiences for whom the words are intended (e.g. different native language, different knowledge of terminology).

The contexts in which the words appear range from full natural-language descriptions, which may contain all the characters in their natural order, supplemented by headings, to applications such as conventional keys in which the characters appear in random order, completely out of the context of their related characters. Intermediate cases are descriptions in which parts are omitted because of: missing data; inapplicable characters; inclusion of only a subset of the characters; or inclusion of only diagnostic attributes. Other example of random order are: lists of 'best' characters in interactive identification;

displaying the attributes of a specimen in the order in which they were entered in interactive identification; displaying diagnostic descriptions in the order in which the characters were added.

Another requirement is an abbreviated form of the character for displaying in applications, such as interactive identification, where characters must be selected from a list. In DELTA, this is achieved by means of comments in the ‘feature’ line of a character. For example, with the character

#10. leaves <presence>/
1. present/
2. absent/

Intkey would display ‘leaves (presence)’ in character-selection lists, but a natural-language description would read (for example) ‘leaves absent’.

It is often possible to meet the requirements of various contexts from a single character list, though doing so may require some compromise – the results for some purposes may not be optimal. For the best results, it may be necessary to have alternative wordings. In the past, we have accommodated this in DELTA simply by having separate character lists, and invoking the appropriate one for different applications. This is inefficient, because a large proportion of the words can usually be used in all applications. We therefore want to move towards a single list, with groups of words flagged for use in different applications or contexts. The same mechanism can be used for different languages, and also in other text such as character notes, and text in ‘item’ descriptions (text characters, and comments associated with attributes).