

INCORPORATING CHECKS AGAINST OBSERVER ERROR INTO IDENTIFICATION KEYS

BY R. W. PAYNE and D. A. PREECE

*Statistics Department, Rothamsted Experimental Station,
Harpenden, Herts. AL5 2JQ, U.K. and
Mathematical Institute, The University, Canterbury,
Kent CT2 7NF, U.K.*

(Received 9 November 1976)

SUMMARY

A method is suggested for checking for errors that may be made when an identification (or diagnostic) key is used. Such errors may arise either from tests being incorrectly used or observed, or because the specimen examined is aberrant or incomplete. The method involves constructing a *check key* for each endpoint of the main key. Use of the appropriate check key confirms or corrects the initial identification.

INTRODUCTION

Consider taxa a – h whose responses + and – to the binary (or dichotomous) tests T1–T6 are as in Table 1. If the tests are not subject to error, i.e. if the responses observed for each taxon are always exactly as in the Table, some of the tests there are unnecessary for identification; indeed only four tests (e.g. T1–T4) are then required. For any taxon, this number of tests can be reduced further by using an identification (or diagnostic) key like that shown diagrammatically in Fig. 1. Tests are applied sequentially, the results of each determining which further test, if any, should be used next. Thus, for example, a specimen of taxon b would be identified by T1+, T2+, T3–. (Such a set of test responses is termed a *branch* of the key, and points of the key where branches join, i.e. where tests are used, are termed *nodes*.) Thus keys provide a very efficient means of identification. However, the lack of redundant tests in a key means that, if an error is made with any of the specified tests, an incorrect identification results. This is illustrated in Fig. 1, where the correct taxon for each endpoint of the key is followed, in brackets, by a list of the taxa that can lead to that endpoint as a result of a single error en route. Within each pair of brackets, the incorrectly identified taxon is followed by the number of the test at which the error was made. Such errors might arise from the test concerned being incorrectly used or observed, or from the occurrence of incomplete or aberrant forms of the taxa. We shall use the term *observer error* to cover all such errors. The key provides no way of detecting these errors, whereas use of the full set of tests might lead to contradictory test results not corresponding to any of the possible taxa.

Various authors have made suggestions for overcoming this deficiency. Osborne (1963) described a method of incorporating extra tests into an identification key to guard against errors made when a particular test is used. For example consider the taxa i – t whose responses to the binary tests T1–T8 are as in Table 2. It is supposed that T1 is particularly difficult with taxa m and n ; thus an error is possible in using T1 with these taxa. Fig. 2

shows a key based on Table 1. When a specimen of taxon *m* or *n* is being identified, a negative response to T1 may be observed by mistake, and the wrong branch taken. However this mistake will be detected by T3, after which the correct branch will be rejoined. This linking of different branches is termed *reticulation*.

Table 1. *The responses of taxa a-h to the binary tests T1-T6*

	T1	T2	T3	T4	T5	T6
<i>a</i>	+	+	+	+	+	+
<i>b</i>	+	+	-	+	-	+
<i>c</i>	+	-	+	-	-	+
<i>d</i>	+	-	-	-	+	-
<i>e</i>	-	+	+	-	-	-
<i>f</i>	-	+	-	+	+	+
<i>g</i>	-	-	+	+	+	+
<i>h</i>	-	-	+	-	+	+

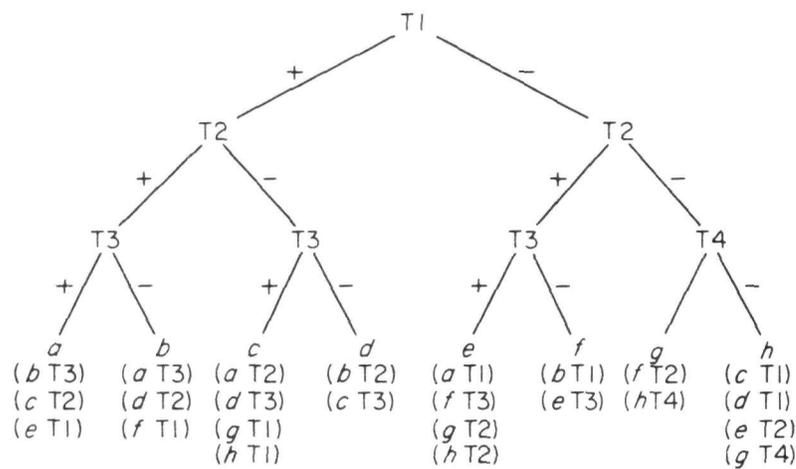


Fig. 1. Diagrammatic representation of a key for taxa *a-h* of Table 1. At each endpoint, the correct taxon is followed by a list of incorrect identifications, each arising from an error with the test indicated.

Table 2. *The responses of taxa i-t to the binary tests T1-T8, test T1 being particularly difficult for taxa m and n*

	T1	T2	T3	T4	T5	T6	T7	T8
<i>i</i>	+	+	+	+	-	+	-	+
<i>j</i>	+	+	-	+	-	+	-	+
<i>k</i>	+	+	-	-	-	+	-	+
<i>l</i>	+	+	-	-	-	-	-	+
<i>m</i>	+(?)	-	+	-	+	+	-	+
<i>n</i>	+(?)	-	+	-	-	+	-	-
<i>o</i>	-	-	-	-	+	+	+	+
<i>p</i>	-	-	-	-	+	-	+	-
<i>q</i>	-	-	-	-	-	+	-	-
<i>r</i>	-	-	-	-	-	+	-	-
<i>s</i>	-	-	-	+	-	-	-	-
<i>t</i>	-	-	-	-	-	-	-	-

Reticulation is not essential in this method. Fig. 3 shows another key in which the results of T1 are checked but here the branches for taxa *m* and *n* do not rejoin. This key also shows that the deliberate selection of tests in order to facilitate reticulation as in Fig. 2 may lead to a less efficient key: in Fig. 2 the average number of tests per identification is $4\frac{1}{6}$, whereas in Fig. 3 only four tests are required per identification even if a mistake is always made when

T1 is used with taxa m and n . Reticulation is none the less a useful device for reducing the size of a key containing duplicate sub-keys. Payne (1977) explains how the facility for reticulation has been provided in GENKEY, a general computer program for constructing identification keys (Payne, 1975a, b).

Thus Osborne's method is equivalent to treating T1 as though it were variable within taxa m and n . In this equivalent form the method has been widely used. For example Barnett (1971), in selecting nutritional tests for identifying yeasts, treated responses that are slow or feeble (and thus difficult to observe) like those that vary from strain to strain within a yeast species. This approach is possible only if not all the responses are subject to error.

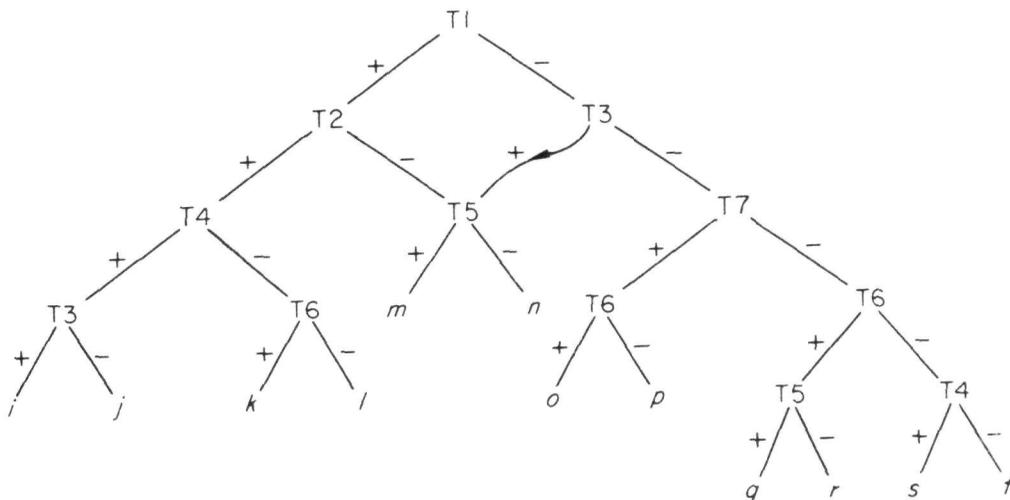


Fig. 2. A key for the taxa $i-t$ of Table 2. An error in using T1 on taxon m or n is detected by T3 and the correct branch is rejoined by means of reticulation.

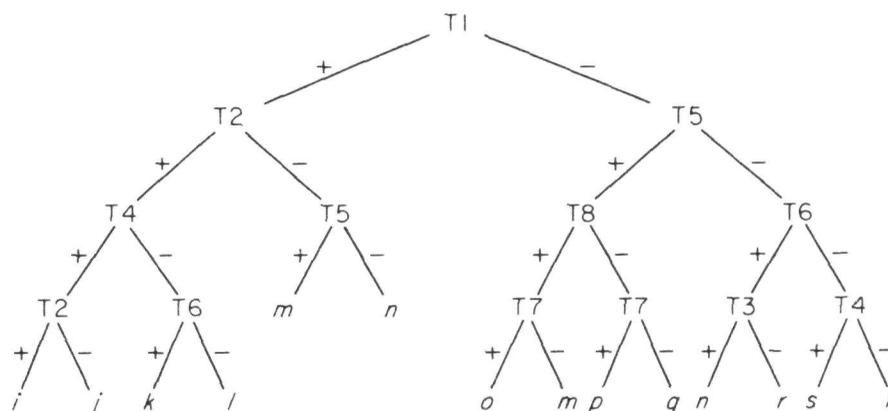


Fig. 3. A further key for taxa $i-t$ of Table 2. The results of T1 on taxa m and n are checked as in Fig. 2 except that reticulation is not used.

Willcox *et al.* (1973), when identifying species of bacteria, regarded all tests as subject to error either in performance or in the transcription of the results. For each possible response to each test for each species, they therefore estimated the probability that a specimen selected at random from that species would give that response, either as a true response or because of an error in applying or observing the test. They also estimated prior probabilities for each species, and used a probabilistic identification method.

A probabilistic key is likely to be considerably larger than a non-probabilistic key with a similar number of taxa, and construction takes longer, even if the approximations suggested by Willcox *et al.* (1973) are used. Thus this method is likely to be useful only if, as with bacteria, most of the test results are already variable as the result of variability within taxa.

Treating variability within taxa in the same way as observer error does, however, give rise to problems. The probability of observer error is likely to vary greatly amongst users of keys. More importantly, a test that is considered variable only as a result of observer error can be checked by repeating the test, provided the testing is non-destructive, to obtain the true result for the taxon concerned. This repetition of tests, which cannot be incorporated in the probabilistic method, is used in the method described below.

THE USE OF EXTRA TESTS FOR CHECKING AN IDENTIFICATION

The method we suggest has two phases. In the first the key is constructed in the usual way. Particularly difficult responses can be treated as variable, as described above, but all other responses are regarded as *fixed* (i.e. non-variable).

Gower and Payne (1975) review methods used in computer programs for key construction. These methods are all similar: the key is constructed sequentially, node by node, the test selected at each node being the one that best subdivides the taxa occurring there. (A taxon is said to *occur* at a given node if specimens of the taxon can give the test results leading to that node.) The test-selection methods use some function of the numbers of taxa giving the different possible responses to the test. The basic aim is to make these numbers as nearly equal as possible while keeping the number of taxa with variable responses small. We discuss below how the selection of tests can be modified to make subsequent checking more efficient.

The key constructed in the first stage is sufficient for correct identification provided the user makes no errors. (If the set of tests is not sufficient to distinguish completely all the taxa, groups of taxa may occur at some of the endpoints.) We shall suppose that errors are very unlikely and indeed that a user will make at most one error per identification. For long or particularly difficult keys, this may not be a reasonable assumption; however, the method can easily be extended to allow for two and more errors.

In the second phase, each endpoint of the key is considered in turn and the set of taxa that could occur there as the result of at most one error is constructed. (Fig. 1 shows that the sets at different endpoints may be of different sizes.) To check for an error, it is sufficient to select additional tests to distinguish between the taxa in the set at the endpoint concerned. This can be achieved either by extending the main key, constructed in phase one, or by constructing a separate *check key* to be used after reaching the endpoint. The latter alternative seems preferable as it allows the confident user to dispense with checks and prevents the main key from becoming too long and cumbersome.

The check key is constructed in three stages. In the first, an initial check key is constructed using the same methods as for the main key except that tests already used on the relevant branch of the main key are now excluded from consideration. (Thus it may not be possible to distinguish all the taxa in the set—even if complete identification was possible with the main key.) Fig. 4 shows an initial check key for the leftmost endpoint of the key in Fig. 1. This check key uses tests T4–T6; although T4 is used in the main key, it is not on the branch to this endpoint.

The endpoints of the initial check key are then scanned. If, as with the leftmost endpoint in Fig. 4, the initial check key confirms the identification in the main key, no further action need be taken; at least two errors would be needed for this identification to be false. However, an endpoint in the initial check key may identify a group of taxa not containing the taxa originally identified in the main key; a user who reaches such an endpoint has made an

error in either the main key or the initial check key. To discover where the error has occurred, any of the original taxa that could reach that endpoint as the result of a single error should be added to the group at that endpoint and the initial check key extended with further tests from those not used already. In the initial check key in Fig. 4 the original taxon, *a*, could reach either of the two middle endpoints as the result of a single error. However, the rightmost endpoint would require errors in both T4 and T6; thus, if that endpoint is reached, identification of taxon *e* is known to be correct. Fig. 5 shows the check key extended with tests not used already.

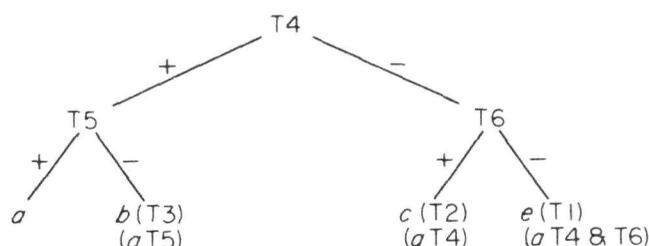


Fig. 4. The initial check key for the leftmost endpoint of the key in Fig. 1.

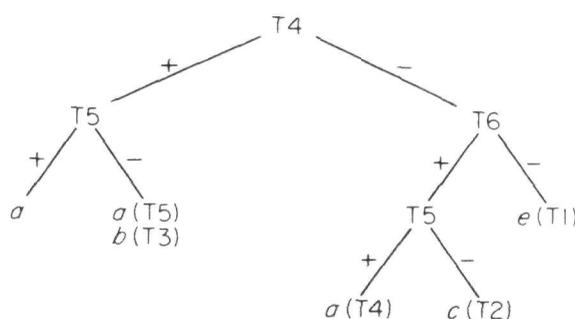


Fig. 5. The check key for the leftmost endpoint of Fig. 1, after the second stage of construction.

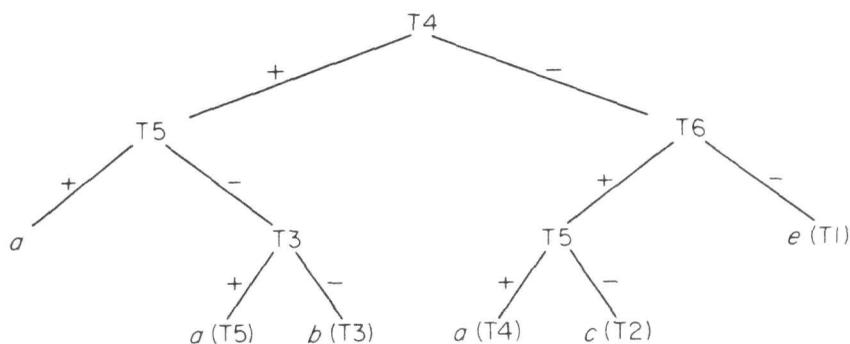


Fig. 6. The completed check key for the leftmost endpoint of Fig. 1.

After the first two stages, the check key distinguishes between the taxa as far as possible without re-using any tests. However, if any test cannot be checked independently, or if, as at the second endpoint of Fig. 5, an error has been detected but cannot be resolved independently, mixtures may occur of the originally identified taxa (which may or may not be erroneous) and other taxa arising from errors. It is always preferable to check tests independently as careless errors are often repeated; however, tests can if necessary be repeated as described above. (Possibly the fact that a test occurs a second time will be sufficient to encourage the user to be especially careful.) Thus, in the final stage, the check key is extended to give as complete an identification as possible with tests already used. Fig. 6 shows the final check key for the leftmost endpoint of Fig. 1. At each endpoint of a check key the test results in the main key and the check key can be compared with those in the table of results for the identified taxon so that the name of any test in which an error has occurred can be printed.

If any of the tests has more than two possible results, some of the endpoints in the main key may be *null*, i.e. no taxa may occur there. This will happen if, for example, a test with three possible results is used to distinguish between two taxa. The user can reach such an endpoint only as the result of an error. Construction of the check key for such an endpoint is thus less complex than for a non-null endpoint. A null endpoint in the final check key can be reached only as the result of two errors, at least one occurring in the check key. Such an endpoint is perhaps best left null to indicate that the specimen concerned requires a more rigorous or detailed examination. Alternatively the null group could be replaced by the group resulting from two errors, and the check key extended.

SELECTION OF TESTS IN THE MAIN KEY

As not all users will wish to make checks, the efficiency of a check key is less important than that of the main key.

Gower and Barnett (1971) describe how secondary considerations, such as whether a test already occurs on another branch of a key, can be used to influence key construction. A selection function is used as before; however, instead of the best test being chosen, a group of *equivalent tests* is formed containing those tests for which the value of the selection function differs from that of the best test by less than a predetermined constant. The test to be used is chosen from this group using some secondary criterion. Secondary criteria that can be used for selecting tests in the main key to make the check keys more efficient overall are described below.

For a given node of the main key, let $\{P\}$ denote the *natural set* of taxa that occur at that node if there are no errors in tests leading to it, and let $\{S\}$ denote the *error set* containing taxa that can occur there as the result of one error. The error sets for subsequent nodes will each consist of some subset of $\{P\}$ plus some subset of $\{S\}$. The sizes of the error sets for subsequent nodes will be reduced if a test can be chosen with more than two possible results, some of which are not given by any taxon in $\{P\}$ but which are given as fixed responses by taxa from $\{S\}$; this adds extra endpoints to the key, but each of these is reached only as the result of an observer error. If there is more than one such test in the group of equivalent tests, the test chosen should be the one for which the most members of $\{S\}$ give, as a fixed response, any of the responses not given by taxa in $\{P\}$.

If there is no such test in the group of equivalent tests, suppose test t from the group has $l(t)$ possible responses which produce natural sets of taxa of sizes $m_1(t), \dots, m_{l(t)}(t)$ and error sets of sizes $n_1(t), \dots, n_{l(t)}(t)$. If the taxa can all be distinguished the test chosen should have ratios $\{n_i(t)\}/\{m_i(t)\}$ as nearly equal as possible. This will tend to make the check keys of approximately equal size, and thus to improve their efficiency overall. If not all the taxa can be distinguished, and the number of taxa in $\{P\}$ is less than some pre-determined constant, the test can be chosen with the quantities $n_i(t)$ as nearly equal as possible. The closeness to one another of the quantities $\{n_i(t)\}/\{m_i(t)\}$ or $n_i(t)$ can be assessed using a selection function similar to the function used to form the group of equivalent tests.

DISCUSSION

The probabilistic approach to guarding against errors made by users of keys incorporates a probability of error into the probability of each response. This means that every user must

check for errors. The approach has the further disadvantage that the user is given no opportunity to repeat a test in which he may have made an error. If the user is assumed to make at most one error, a subsidiary check key to correct or confirm the initial identification can be constructed using as few as possible of the original tests. This method can be extended in an obvious way to check for more than one error, although this will make the check key rather larger.

The construction of check keys has been implemented in the computer program GENKEY (Payne, 1975a, b). Further information on facilities and availability of GENKEY can be obtained from the first author.

REFERENCES

- BARNETT, J. A. (1971). Selection of tests for identifying yeasts. *Nature New Biology*, **232**, 221.
- GOWER, J. C. & BARNETT, J. A. (1971). Selecting tests in diagnostic keys with unknown responses. *Nature*, **232**, 491.
- GOWER, J. C. & PAYNE, R. W. (1975). A comparison of different criteria for selecting binary tests in diagnostic keys. *Biometrika*, **62**, 665.
- OSBORNE, D. V. (1963). Some aspects of the theory of dichotomous keys. *New Phytol.*, **62**, 144.
- PAYNE, R. W. (1975a). Genkey: a program for constructing diagnostic keys. In: *Biological Identification with Computers* (Ed. by R. J. Pankhurst), pp. 65–72. Academic Press, London.
- PAYNE, R. W. (1975b). *Genkey*. Inter-University/Research Councils Series. Report No. 24. Program Library Unit, Univ. of Edinburgh.
- PAYNE, R. W. (1977). Reticulation and other methods of reducing the size of printed diagnostic keys. *J. gen. Microbiol.* **98**, 595–597.
- PAYNE, R. W. (1977). Reticulation and other methods of reducing the size of printed diagnostic keys. *J. gen. Microbiol.* **98**, 595–597.
- WILCOX, W. R., LAPAGE, S. P., BASCOMB, S. & CURTIS, M. A. (1973). Identification of bacteria by computer: theory and programming. *J. gen. Microbiol.*, **77**, 317.

This document is a scanned copy of a printed document. No warranty is given about the accuracy of the copy. Users should refer to the original published version of the material.