CAFCA

Chapter 3

Group Compatibility
-
Primary Analysis

February 1996

THIS PAGE INTENTIONALLY LEFT BLANK

III. GROUP COMPATIBILITY - PRIMARY ANALYSIS

INTRODUCTION

CAFCA employs the methods of group-compatibility (Zandee and Geesink, 1987) and component-compatibility (Zandee and Roos, 1987) to run its cladistic analyses. In contrast to, say, standard parsimony methods CAFCA does not find cladograms by exploring a search space looking for cladograms that maximize an optimality criterion expressing a property of a cladogram in terms of the characters used, like cladogram length. Instead, CAFCA explores a search space looking for cladograms with maximum resolution given the components available. The components in their turn can be defined in terms of character states in various ways. The collection of cladograms found can be delimited further by applying different optimality criteria.

This manual introduces you to these methods by taking as a starting point the results that can be obtained by applying them. In this way it is possible to guide you one step at the time, from a straightforward type of analysis to the more complicated ones.

CAFCA output has 6 parts.

1. a Header, including CAFCA parameter settings
2. the Data Matrix.
3. a list of Building blocks for cladograms (clada, components), and their corresponding character states.
4. a list of Character states on the root of selected cladogram(s).
5. a list of Selection criteria for cladograms.
6. Diagrams of selected cladograms, plus their lists of apomorphies and character state changes.

You will get a guided tour along these items, using a simple example of a data matrix with 5 taxa, Aus, Bus, Cus, Dus, and Eus, and 8 characters. The data matrix is available in the examples folder on your distribution disk.

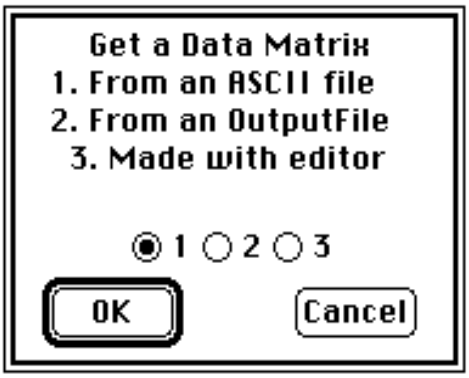
EXAMPLE

The first analysis will be very straightforward, using the program's defaults for the major parameters. The data matrix used is complex enough to introduce the several possibilities of CAFCA as to coding of characters in a binary matrix, but on the other hand also simple enough from the point of phylogenetic structure as not to allow many competing best cladograms. The same data matrix will be used in the next examples as well, to illustrate the effect of changing some of the parameters.

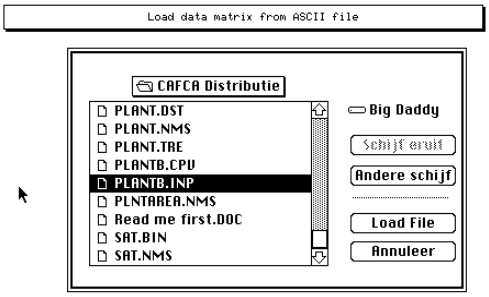
TUTORIAL

1. Select **Primary Analysis** in the **Run** menu.

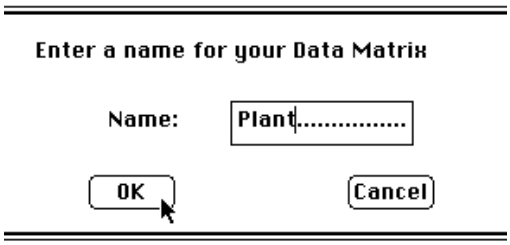
2. Click **1** (**From an ASCII file**) and **OK** in the dialog.



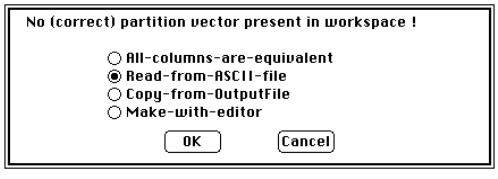
3. Select **PLANTB.INP** from the example data on your distribution disk and click **Load File**.



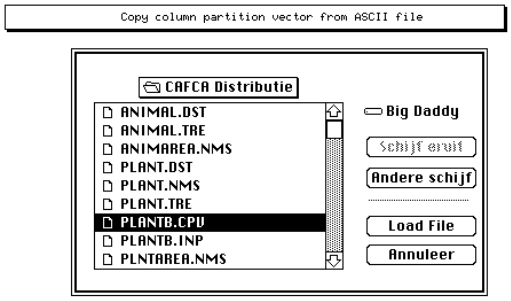
4. Enter '**Plant**' (without quotation marks!), for example, as a name for your data matrix in the next dialog box.



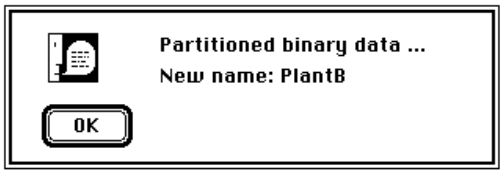
5. Click **Read-from-ASCII-file** in the **No (correct) partition vector present** dialog box.



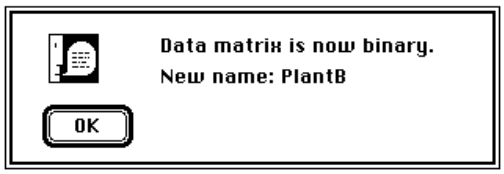
6. In the next file selector box select **PLANTB.CPV** from the example data on your distribution disk and click **Load File** .



Nota Bene: CAFCA runs it analyses on binary (0/1) expressions of the data matrix. If the data-file (step **3**) already contains binary expressions of the multi-state characters in the data matrix, as in this case, CAFCA computes the multi-state expression of the data by means of the partition vector and renames the binary expression by adding the letter 'ΔB" to the name given previously (step **4**). CAFCA notifies you with the following message box.



If, on the other hand, your data matrix contains multi-state characters and already has a multi-state expression when read in (as is the case with the file **PLANT.ASC** in the examples folder) than step **6** will be skipped as CAFCA can now compute the partition vector from the data matrix itself. You will get the following message box instead.



7. Click **No** in the dialog presenting you the option for separate columns for character states in polytypic taxa .

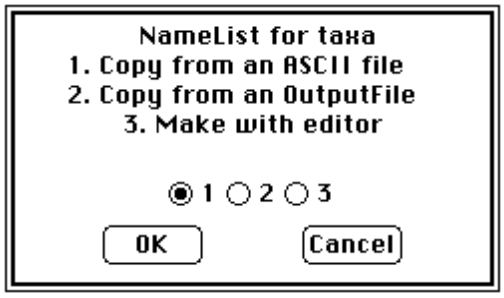
Nota Bene: If the ASCII file with your data matrix already contains names for your taxa, like in the following example,

```
/ This is an example of an ASCII
  file containing a data matrix,
/ representing the distribution
  of character states
```

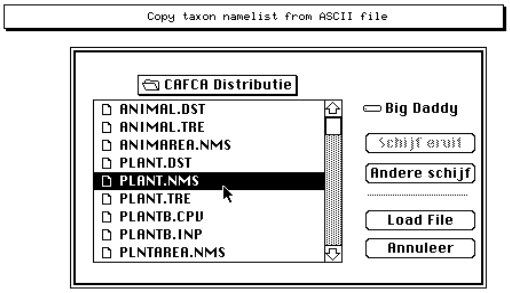
```
/ over all terminal taxa.
Aus
2020110011
Bus
1120110011
Cus
1010110021
Dus
1020121111
Eus
1120031111
" You may start the inputfiles
  for CAFCA with comments,
" i.e. text preceded by either a
  # / \ * " ' ; or !
' and you may add closing com-
  ments as well, like this one.
/ You can add names for the ter-
  minal taxa.
/ Names for taxa may contain
  {}()[]-+._
/ and integers 01234567789 when
  enclosed by alphabetic charac-
  ters.
/ Spaces in names are not al-
  lowed and replaced by _ when
  reading inputfiles.
```

then CAFCA will not bother you with steps **8** and **9**, as the names for the taxa are extracted from the input file, together with the data matrix.

8. Click button **1** (**Copy from an ASCII file)** in the **Namelist for taxa** dialog box.

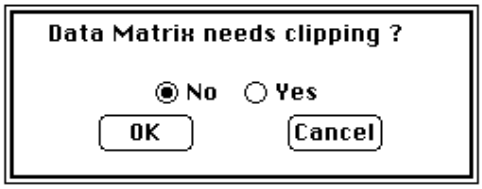


9. Select **PLANT.NMS** from the example data on your distribution disk and click **Load File**.

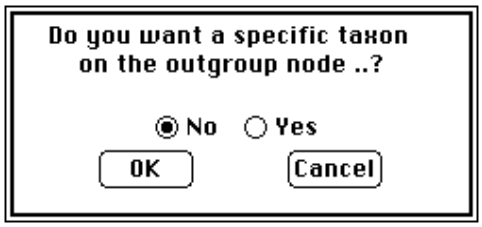


10. Click **No** in the **Do you want to edit your data** dialog.

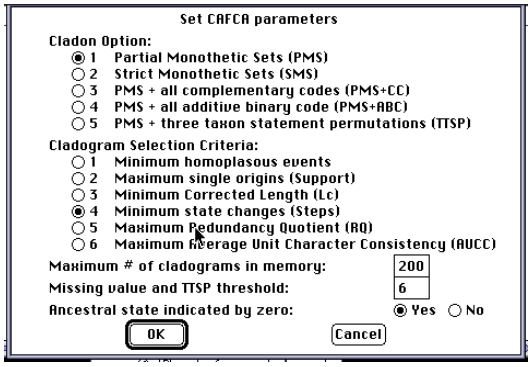
11. Click **No** in the **Data matrix needs clipping ?** dialog box.



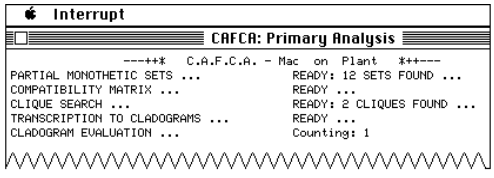
12. Click **No** in the **Taxon on the outgroup node** dialog box.



13. Take all defaults in the **Set CAFCA Parameters** dialog box. Then click **OK**.

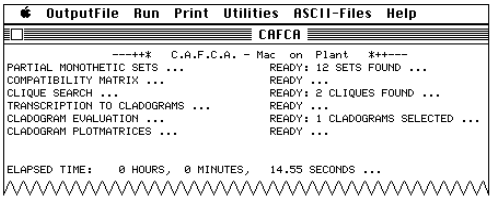


The analysis will now start running. Its progress can be followed on the screen..

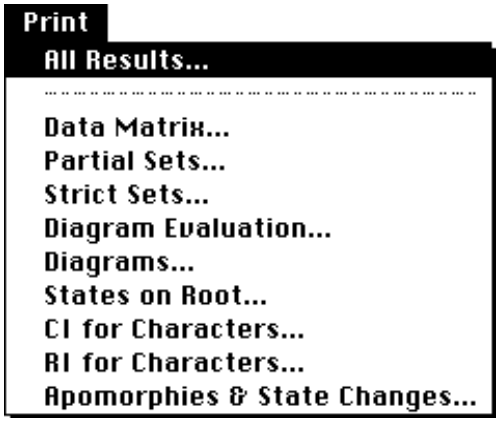


If for whatever reason you want to stop this run, use the **Interrupt** menu (see

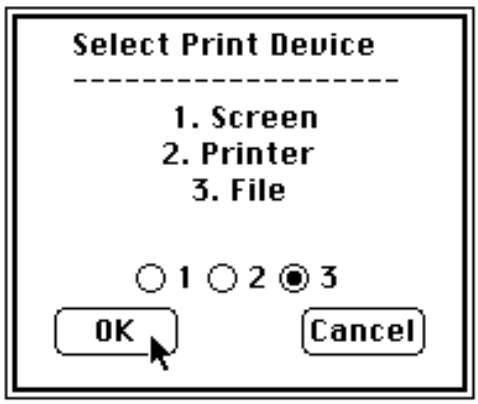
chapter 2). When the elapsed time message appears the analysis is finished



14. Now select **All Results** from the **Print** menu.

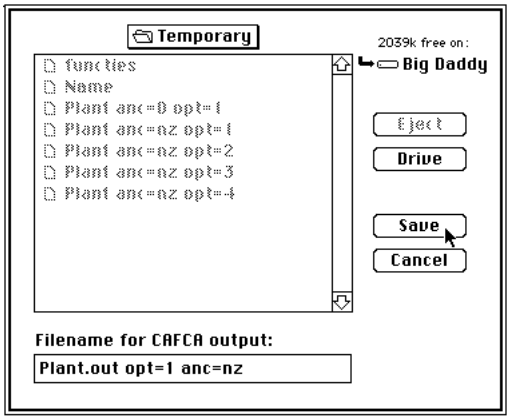


15. Sending a file to a (appletalk connected) laserprinter from within CAFCA is now reasonably up to standards.

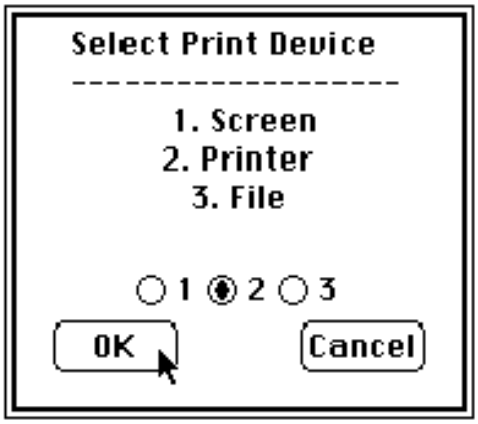


So, you can either print your data to file, and use your favourite word processor to open, edit and print the output of a CAFCA run, or you could print from within CAFCA. If you choose the first option you should click button **3** (**File**) and **OK** in the dialog.

16. In the following file save box you can enter a name for the file where the output will be written to.

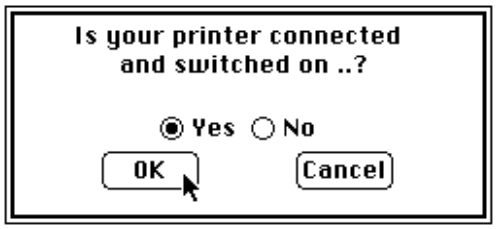


17. If, however, you do want to send your output to a printer directly, click button **2** (**Printer**) in the **Select Print Device** dialog box.



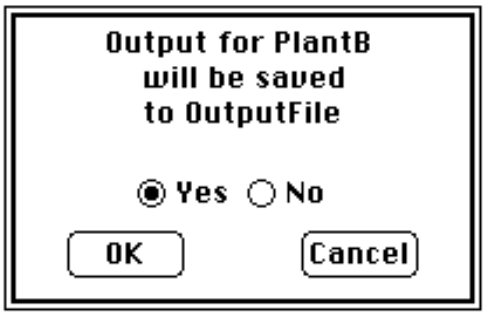
18. Click the printer of your choice in the **Select Printer** dialog box. The default choice, **any printer**, will suffice for any (appletalk) connected printer. The other two options are in fact relicts of my own printing facilities, which needed character translation to function properly.

19. Click **OK** in next dialog if everything is all right with your printer.

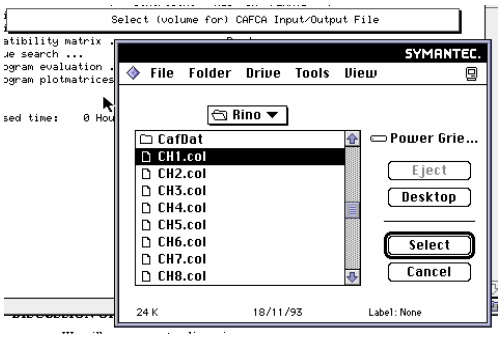


20. After printing your results on a printer or to file, select **Save & Resume** in the **Output-File** menu if you want to save your results in a file, to be used later on, eventually.

21. Click **OK** in the **Output for PlantB will be saved to OutputFile** dialog box if you really want to save your results .

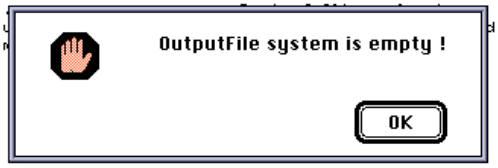


22. CAFCA will continue by asking you to select a volume and folder in which to save the outputfile.

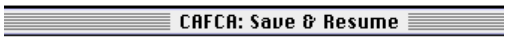


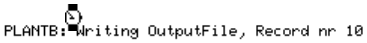
If you have never saved any data to an OutputFile before, you can click any file within the folder of your choice. CAFCA won't use this file but it will use this folder to write its output to.

23. CAFCA will notify you of its discovery that there is no OutputFile system present yet in this folder by means of the following dialog. Click **OK**.



24. CAFCA will start writing the output of this primary analysis of PLANTB to its OutputFile. The results of several analyses can be written to the same OutputFile. This file can be recognised by the name CAFCA.IO The results of the separate analyses can be found and retrieved from this file-system by their own name, PLANTB in this particular case.





DISCUSSION OF RESULTS

I will now present a discussion of the results obtained in this first primary analysis, and at the same time explain the concepts, jargon, and peculiarities of CAFCA.

```
          CAFCA - Mac   version 1.3i  (c) M.Z. 1987, 1995
          Date : 1 AUG 1995     Time : 9H30M28S

                    CAFCA Parameter Settings
Type of analysis ..................... Primary
```

```
Cladon option ......................... 1: Partial  Monothetic Sets (PMS)
Cladogram Selection Criterion ......... Minimum Length
Taxon on outgroup-node ................
Ancestral zero's in character ......... 4 5 10
Ordered characters .................... 10
Maximum Number of Cladograms .......... 2
```

Table 3.1 Header of CAFCA output of first example.

HEADER

The header section (table 3.1) recapitulates the values of the major CAFCA parameters, as set before the analysis, either by you or by the program (defaults).

Type of analysis.

Type of analysis indicates your choice for either a

- ➢ primary analysis,
- ➢ secondary analysis,
- ➢ biogeographic analysis,
- ➢ user-tree evaluation.

Primary analyses are run with the complete character data matrix as a source of building-blocks (clada) for cladograms. These building blocks are defined as monothetic sets (Beckner 1959; partially or strict). Their number can be increased, optionally, by including all additive binary codings for each multistate character. By using this option you check every possible sequence of states as a hypothesis of homology. You may, as an option, also include all groups from valid three-taxon-statement permutations to increase the number of building blocks for cladograms. A three-taxon-statement is considered valid if each cladon (group of terminal taxa) within the statement is supported by its own independent (local) synapomorphy.

Cladograms for taxa are derived as general patterns of internested (hierarchical) groups of taxa, emerging from the combination of the particular (independent) pattern in each separate character.

Secondary analyses serve to resolve polytomies in cladograms resulting from a previous primary or biogeographical analysis or user-tree evaluation where the set of building blocks for cladograms contained insufficient information for complete dichotomous resolutions (see chapter **4**).

Biogeographic analyses are run to explore the relations between the phylogeny of a group of taxa, or different phylogenies of different (unrelated) groups, and the geographical distribution of the taxa involved (see chapter **6**).

This type of analysis can also be used to explore the historical relationships between parasites and hosts, by considering hosts as areas of endemism for parasites. In fact any co-evolutionary pattern can be studied for its historical implications by this type of analysis. You may even consider taxa as areas of endemism for genes (character state expressions). The phylogeny of the taxa is seen as the general pattern emerging from the separate phylogenies of independent genes (character carriers), just as a general pattern for the historical relations among areas of endemism emerges from the separate phylogenies of independent taxa. That’s the reason why in CAFCA primary, secondary, and biogeographic analyses are **identical** as to the method employed.

User-tree evaluation takes place when you have entered a data matrix plus one or more cladograms that must be evaluated against this data matrix. The cladograms usually come from the literature, or are based on intuition, but are as a rule not directly derived from the data matrix itself (see also chapter **5**).

User-trees need not be completely resolved. If they are unresolved (i.e., contain polytomies) they can be subjected to a secondary analysis after evaluation.

Another possible use of user-tree evaluation may result from running a primary analysis on a data matrix containing the 'better' characters that, however, do not give a completely resolved cladogram. After saving, this cladogram can be entered as a user-tree and evaluated against another data matrix containing the 'weaker' characters, and consequently subjected to a secondary analysis on the basis of the 'weaker' characters.

User-tree evaluation is also applied in co-evolutionary studies. In those cases an independent estimate of the host phylogeny may be available. This host phylogeny is evaluated against the cladogram(s) for hosts found from the data matrix based on the parasite phylogeny and the distribution of parasites over hosts.

Cladon option.

The cladon option refers to the way the building blocks for cladograms (clada) are defined. In the first example these clada are defined following the partial definition for monothetic sets.

Cladogram selection.

In CAFCA you can choose from six different cladogram selection criteria. In the first example the default option, cladogram length, is chosen.

Ancestors and outgroups.

CAFCA can be run without an outgroup being indicated in the data matrix. On the other hand an outgroup can be declared optionally and interactively by you, or the program may deduce an outgroup from the presence of a full-zero row in the data matrix.

In a multi-state character a zero entry is interpreted as an indication of a (putative) ancestral state (see the paragraphs on 'assumptions regarding zero's in the data matrix').

Ordering of (multi-state) characters.

All characters are treated as unordered as well as unpolarized (unless ancestral zero is present and indicated as such) by default, with the exception of characters with (incomplete) additive binary coded states in binary data matrices, and multi-state characters for which you implemented a linear ordering upon request by the program.

In case a multi-state data matrix is used as input, the program will ask if these characters (none, some, all; if some then which) should be treated as ordered, that is, seen as an a priori polarised and ordered sequence of states. CAFCA can order multi-state characters only linearly in a sorted sequence (0 -> 1 -> 2 -> 3, etc...). Thus if you want the states ordered like 2 -> 1 -> 3 -> 0 you should first renumber them to 0, 1, 2, and 3, respectively.

Binary characters (0/1) are seen as characters with only one state (1) as CAFCA groups taxa as a result of presence (= 1) of states only (see also the assumptions regarding zero's in the data matrix). This state should best indicate a putative apomorphy if true phylogenetic results are required If such putative

polarisation is impossible or unwarranted you should transform the binary character (0/1) to a multi-state one (1/2), ***or*** you should click **No** for 'Ancestral state indicated by zero' in the CAFCA parameter dialog. In the latter case groups of terminal taxa will also be based on the distribution of zero's as these zero's may now represent an apomorphic state.

So, if you want to implement an *a priori* polarisation plus ordering of character states you should either apply binary matrices with (incomplete) additive binary coding, or enter a multi-state data matrix and enforce a linear ordering for all or some of the characters when the program prompts you to do so.

In the present example character 10 shows a linear ordering of its states in the binary image of the data matrix (table 3.2).

Number of cladograms.

In the CAFCA parameter dialog box you can declare what the maximum number of cladograms (MNC) should be for which results will be retained in memory. In the header this parameter is represented by its declared value. Note that this number is different from the maximum number of cliques of components that CAFCA stores during its clique search (a built-in maximum of 5000).

DATA MATRIX

Types of characters, and the partitioning of columns.

CAFCA accepts both binary (0/1), multi-state (0/1/2/3/etc..) and mixed binary/multi-state matrices as input (table 3.2). In the latter case the binary characters are restricted to two states only (0/1). You can not mix the multi-state (0/1/2/3/etc.) representation and the binary representation of multi-state characters in one and the same data matrix.

Character states must be represented by digits (integers). Other symbols, like items from the alphabet, are not allowed. Missing values are allowed and must be indicated by a negative integer or a question mark.

In the analysis a binary representation of the data matrix is used for almost all computations (the cladogram optimisation algorithm uses a multi-state representation). This implies that if you define a multi-state matrix as input a copy of this matrix will be converted into a binary image (with the postfix ΔB added to its name).

The elements of the column partitioning vector (CPV) indicate how the columns (character states) of the binary data matrix should be taken together blockwise. Each block of columns corresponds to 1 character, i.e., a transformation series (= 1 column in the multi-state data matrix); each column in a block represents one character state.

This procedure of treating the binary representation of multi-state characters as blocks of interdependent states avoids the errors that are introduced when each state of a multi-state character is treated as a separate nominal variable (Pimentel and Riggins, 1987).

If you define a binary data matrix as input, you will be prompted to provide a column partitioning vector to let the program know how the character states (columns) should be grouped together, successively, to derive a multi-state matrix. If you enter a multi-state data matrix as input, the program can derive a column partitioning vector for the binary image.

Data Matrix (binary) : PLANT (Columns represent character states)

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Aus	1	1	0	0	1	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0
Bus	1	0	1	1	0	1	0	1	0	0	1	0	0	0	1	0	1	1	0	0
Cus	1	0	1	1	0	0	1	0	1	0	0	1	0	0	1	0	1	1	1	0
Dus	1	0	1	1	0	0	1	0	1	0	0	1	0	1	0	1	1	1	1	1
Eus	1	0	1	1	0	0	1	0	0	1	0	1	0	1	0	1	0	1	1	1

Column Partitioning Vector : 1 2 2 2 3 3 1 1 2 3

Data Matrix (multi-state) : PLANT (Columns represent characters)

	1	2	3	4	5	6	7	8	9	10
Aus	1	1	2	0	0	3	0	1	2	0
Bus	1	2	1	1	1	1	0	1	2	1
Cus	1	2	1	2	2	2	0	1	2	2
Dus	1	2	1	2	2	2	1	0	3	3
Eus	1	2	1	2	3	2	1	0	1	3

Table 3.2 Data matrix for a cladistic character analysis.

The data matrix of our first example (table 3.2) was copied from an ASCII file as a binary data matrix (PLANTB.INP from the Xmpls folder on your distribution disk). There are 3 characters with only 1 state (# 1, 7, and 8), 4 characters with two states (# 2, 3, 4, and 9), and 3 characters with 3 states (# 5, 6, and 10).

Apparently there is a contradiction between some elements in the CPV (# 4, 5, 9) and the actual number of states in the characters. However, a zero in a multi-state character (like in # 4 and 5) is not treated as a separate state but as an indication of a (putative) ancestral condition (see assumptions regarding zero's). In character 9 state 3 in the multi-state matrix actually reflects a polytypism (see below) for state 1 and 2 in taxon 4 (columns 16 and 17 in binary image).

Note character 10 which in its binary image is additively coded (= a priori polarised and ordered). If you want to implement a priori polarisation plus ordering of character states in a particular character you should apply a binary matrix with (incomplete) additive coding in the block for that character, or enter a multi-state data matrix and enforce a linear ordering for all or some of the characters when the program asks you to do so.

In the case of characters with only one state (characters 1, 7, and 8) this state should best indicate a putative apomorphy if true phylogenetic results are required as CAFCA groups taxa as a result of presence of states only. If such putative polarisation is unwarranted you should transform the binary character (0/1) to a multi-state one (1/2), *or* you should click **No** for 'Ancestral state indicated by zero' in the CAFCA parameter dialog. In the latter case, groups of terminal taxa will also be based on the distribution of zero's as these zero's may now represent an apomorphic state. In all other cases characters are treated as unordered and unpolarized (characters 2 - 6, and 9), unless all states in a block are (incompletely) additive binary coded (character 10; columns 18, 19, 20 in binary image).

Polytypism.

Polytypism for character states in one or more taxa can be coded in different ways. In a multi-state data matrix one can enter a polytypic state with its own code of which, like all other codes, you know what it represents. In a binary data matrix polytypism can be indicated by simply entering a 1 for all the states (within a block of homologous states) present in a taxon. The program then offers you the opportunity either to let the program insert separate (new) columns for each polytypism, or to leave the matrix as it is. In the latter case

polytypisms will show up in the multi-state data matrix by their own code (as they will in the state change list), although in the binary image of the data no separate columns for polytypism are present. CAFCA does not accept codes like {123} or (1,3~5), enclosed either in curly braces or parentheses, as used in *PAUP*'s or *MacClade*'s (both version 3.0) NEXUS file format, to indicate possible assignments ('uncertainty' vs 'polymorphism') of character states.

In the present example taxon 4 shows polytypism for character 9. No separate column for this state is present in the binary data matrix. The multi-state image of the binary matrix, however, shows a distinct code (3) for this polytypism that can be traced as such in the state change list (see page 45). Note, however, that CAFCA has no provision to deal with polytypism in internal nodes (hypothetical ancestral taxa) of the cladogram. It simply does a most parsimonious assignment, according to accelerated transformation (ACCTRAN) of a character state to an internal node.

Assumptions regarding zero's in the data matrix.

1. In a data matrix with multi-state characters, zero's will, as a default, be interpreted as indications of putative ancestral states, except in full-zero columns (full-zero columns are neglected). In the cladogram optimisation process these putative ancestral conditions will be **forced** to be present at the root, even at the cost of extra steps.
If you want a more liberate (and sometimes more parsimonious) attitude towards such putative ancestral states during cladogram optimisation, these states should be given either a question mark or their own code (any number but zero) in the data matrix, *or* you should click **No** for 'Ancestral state indicated by zero' in the CAFCA parameter dialog.

2. In a binary data matrix where columns represent character states and blocksizes for columns indicate the number of homologues represented by contiguous columns, zero's are parameters of a character state indicating a P*i*=0 for finding that state in taxon *i*, unless all entries in a row within a block are zero; then a putative ancestral condition is indicated (= neither one of the states in the block is present in the ancestor). The zero condition is forced to be present at the root in the cladogram optimisation process, unless you clicked **No** for 'Ancestral state indicated by zero' in the CAFCA parameter dialog.
In true binary characters (with a blocksize equal to one !) the number of homologues is actually 1 under the condition that the ancestral state is indicated by zero, and 2 when it is not. In the latter case the zero's (0) in such a true binary character will, like the one's (1) be treated as indicating a group of terminal taxa (cladon), as they may represent an apomorpic state.

In the present example character 4, 5, and 10 show the first assumption in the multi-state image, and the second assumption in the binary image. The second assumption is also demonstrated by characters 1, 7, and 8, although a full-zero row is not present here. For the other characters 2, 3, 6, and 9 (without zero's) these assumptions do not obtain.

Full-zero columns

Full-zero columns in either a multi-state or a binary data matrix will pass the analysis as dummy's, i.e., they do not influence any of the results in any way. As a weight option is not yet implemented in CAFCA, this characteristic gives you a provisional opportunity to run different analyses on different selec-

tions of characters by substituting zero's for some of the characters, without the need of changing the size of the data matrix and thereby the index numbers of the characters. When you use the **Clip data matrix** option in the **Utilities** menu or during the preparation of a (primary) analysis, this is what happens with columns that are marked for deletion by you.

Missing values.

Missing values are allowed and must be indicated by a negative integer. For a binary data matrix this implies that missing values are represented by -1. In a multi-state data matrix either -1 or any other negative integer can be used. In the binary image of a multi-state data matrix negative integers show up as -1 in the appropriate column in the respective block. In a multi-state data matrix it is allowed to indicate a missing value for one taxon by, say, -2, and another missing value in the same character but for another taxon by, say -4, suggesting that these states are not alike, although unknown.

For identical indicators of missing values for several taxa, say, three taxa all showing a -2, all possible combinations of these taxa with the taxa showing known states with value 2 will be used in the derivation of building-blocks for cladograms. This is likewise true for taxa showing -1 with those showing 1 as a known state, those showing -3 with those showing 3, etc... This procedure implies that a data matrix with one column indicating identical missing values (e.g., -1) for all taxa will result in all cladograms possible given the number of taxa (the default is 6 taxa, implying 945 cladograms; you can indicate otherwise in the CAFCA parameter dialog. The maximum number possible is 12, although using it is quite absurd when you realise the number of cladograms (13.749.310.575) that are implied by this number. You may tie up CAFCA for years.

Thus if you know nothing, i.e., you have no data on your taxa, all possible outcomes are equally likely and will be presented (within limits).

In ASCII files representing data matrices you can also use a question mark as an indicator of a missing value. When CAFCA imports these files the question marks are translated to -1.

CLADA

Building-blocks for cladograms, or clada (singular: cladon), are derived from the binary representation of the data matrix by defining either partial or strict monothetic sets (Beckner, 1959) of terminal taxa, and their corresponding sets of character states (table 3.3a, table 3.8), or by adding to the partial sets the clada resulting from all additive binary codings of all multi state characters (table 3.11), or by adding to the partial sets the clada resulting from all valid three-taxon-statements obtained from all three-taxon-statement permutations (table 3.15).

Although using monothetic sets and variations thereof in the recognition of building blocks for cladograms the group- and component compatibility method should not be confused with the so-called monothetic group method as discussed by Farris, Kluge and Mickevich (1982). In contrast to the latter method CAFCA does not depend on *a priori* specification of transformation series and polarities of characters.

Option 1: Partial monothetic sets (PMS)

Partial monothetic sets of terminal taxa are defined by sets of unique character states (= partial application of the definition for monothetic sets according to Beckner, 1959). Partial monothetic defined clada reflect, as it were, a strong

belief by you in your conjectures of homology, e.g., the branched hairs underneath the leaves are ‘the same’ in all taxa observed (compare dePinna's [1991] primary homologies).

Given a multi state and a binary character for the taxa A to H, like for instance

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

the unordered representation in the binary data matrix will be the following blocks of character states:

```
     1¹ 1² 1³  2¹
A    1  0  0   1
B    1  0  0   0
C    0  1  0   1
D    0  0  1   0
E    0  0  1   0
F    0  1  0   1
G    1  0  0   1
H    0  1  0   1
```

from which, assuming that zero in the binary character implies an ancestral condition, the following list of clada is derived:

ABG, CFH, DE, and ACFGH;

from character states 1^1, 1^2, 1^3, and 2^1, respectively.

If the apomorphy decision for 1 or zero in binary character #2 is still undecided, the list of clada is supplemented by set {BDE} based on character state 2^0.

Option 2: Strict monothetic sets (**SMS**)

Strict monothetic sets are defined by unique combinations of character states (neither one of the separate states need to be unique = strict application of the definition of monothetic sets; see also Sharrock and Felsenstein, 1975; Farris, 1978; Farris, Kluge and Mickevich, 1982)).

Strict monothetic defined clada reflect the first signs of doubt as to the homology conjectures implied by partial monothetic sets. Strict sets say, as it were, that if the initial conjectures of homology are doubtful than a first hint of how these homologies may be broken down is given by the distribution (over taxa) of other states from other characters (congruence).

Given binary characters for the taxa A to H, like for instance

```
     1 2 3 4 (characters)
A    1 1 1 0
B    1 0 1 0
C    1 1 0 0
D    0 0 1 1
E    1 0 0 0
F    0 1 0 0
G    1 1 0 1
H    0 1 0 1
```

the following list of clada is generated under option 1 (PMS):

ABD, DGH, ABCEG, and ACFGH;

from character 3, 4, 1, and 2, respectively (zero's assumed to indicate the ancestral condition).

Using option 2 (SMS) the following clada are generated as well

AB, GH, and ACG.

AB results from the interaction among characters 1 and 3 (if ABD is not based on a character state, homologous over taxa, then maybe AB is), GH from characters 2 and 4, and ACG from characters 1 and 2. In this way we are still limiting the number of homoplasies to account for if character state distributions are not fully congruent. For instance, the set GH is not broken down in G and H separately unless there is evidence supplied by other characters that we should do so.

As we should not burden our analysis with hypotheses of homoplasy beyond necessity (Hennig's auxiliary principle), we usually take partial monothetic sets for clada (option 1) as first approximations in our attempts to achieve a fully resolved and parsimonious explanation of our data in terms of a cladogram. This approximation can be made better, if need be, by using strict monothetic sets (option 2) in another attempt to achieve fully resolved most parsimonious cladograms.

Option 3: PMS + all complementary codes

In defining sets of taxa by unique character states only, one may miss MPC's for which it is necessary to assume reversal(s) in order to fit character state distribution(s) to a cladogram. Using PMS + all complementary codes may serve as a first approximation to repair this anomaly, if necessary.

Instead of breaking down clada into subsets as indicated by overlapping character states as **SMS** does, this option finds new clada by iteratively joining the distributions of pairwise overlapping character states. For instance the binary data used in the example above (option 2) to derive strict monothetic sets result in the following sets after the first iteration.

{ABCDEFGH} <> 1 + 2
{ABCDEG} <> 1 + 3
{ABCDEGH} <> 1 + 4
{ABCDFGH} <> 2 + 3
{ACDFGH} <> 2 + 4
{ABDGH} <> 3 + 4

In the second iteration these joint distributions are combined among each other as well as with the distributions of the original character states, e.g., {1 + 2} will be combined with {3} and with {4}, as well as with {3 + 4), etc. The iterations stop until no new combinations of sets are found.

This option # 3 can also be applied to strict monothetic sets. If you chose option 2 (SMS) right from the beginning, you will be prompted by CAFCA whether you want to add the complementary codes to the SMS's as well.

Option 4: PMS + all additive binary codes (PMS + ABC).

This option only applies to multi-state characters in a data matrix. For these characters the following algorithm is used to derive clada in addition to those obtained by PMS (option 1).

Given a multistate character for the taxa A to H, like for instance

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

the unordered representation in the binary data matrix will be the following block of character states:

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

from which the following permutations of additive binary codings (trans formation series) are derived

	1⇒2⇒3	**1⇒3⇒2**	**2⇒1⇒3**	**2⇒3⇒1**	**3⇒2⇒1**	**3⇒1⇒2 (series)**
	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3	1 2 3(states)
A	1 0 0	1 0 0	1 1 0	1 1 1	1 1 1	1 0 1
B	1 0 0	1 0 0	1 1 0	1 1 1	1 1 1	1 0 1
C	1 1 0	1 1 1	0 1 0	0 1 0	0 1 1	1 1 1
D	1 1 1	1 0 1	1 1 1	0 1 1	0 0 1	0 0 1
E	1 1 1	1 0 1	1 1 1	0 1 1	0 0 1	0 0 1
F	1 1 0	1 1 1	0 1 0	0 1 0	0 1 1	1 1 1
G	1 0 0	1 0 0	1 1 0	1 1 1	1 1 1	1 0 1
H	1 1 0	1 1 1	0 1 0	0 1 0	0 1 1	1 1 1

as well as the codes for the branched varieties 3⇐1⇒2, 1⇐2⇒3, and 1⇐3⇒2.

These binary codes are used to derive partial monothetic sets, as shown under option 1 above, in addition to the sets already obtained from the original binary codes of the character states in the data matrix. Thus the list already obtained in option 1, ABG, CFH, DE, and ACFGH, is supplemented with the clada ABCDEFGH, CDEFH, ABDE, AB, ABDEG, and ABCFGH.

Option 5: PMS + Three-Taxon-Statement permutations (PMS + TTSP).

This option applies to binary as well as to multi-state characters. For these characters the following algorithm is used to derive clada in addition to those obtained by PMS (option 1). It is based on an unpublished MS (M. Zandee - Three taxon statement permutations + outgroup comparison = cladistic analysis. Paper read at the 4th meeting of the Willi Hennig Society, 1984).

1. Take the binary representation of a character state, e.g.

```
A    0
B    1
C    1
D    1
E    0
F    0
G    1
H    0
```

if this representation contains six 1's or less (6 as a default; you can indicate more if you want to in the CAFCA parameter dialog).

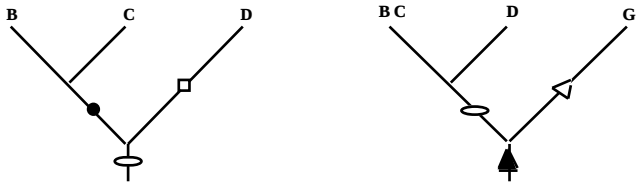
2. Make all groupings of taxa (duo's, trio's, quartet's, etc...) based on the distribution of state present indication (1), e.g.,

BC, BD, BG, CD, CG, DG, BCD, BCG, BDG, CDG, and BCDG

3. Make all possible three taxon statements, based on these groupings, e.g.,
(BC)D, (BD)C, B(CD), (BC)G, (BG)C, B(GC), (BD)G,
(BG)D, B(GD), (CD)G, (CG)D, C(GD), (BCD), (BCD)G,
(BCG)D, (BDG)C, and (CDG)B

4. Check for each three taxon statement whether its constituent parts have any independent character-state support.

Thus, for instance, the groups BC, D, and BCD in (BC)D must have independent (= not identical) supporting character states for the three taxon statement to be considered valid (all characters in the data matrix are used to this end). But the same must be true for BCD, G, and BCDG in (BCD)G, etc...



5. Collect the valid three-taxon-statements.

6. Extract their constituent sets.

7. Do this for the binary representation(s) of each character in the data matrix.

8. Join the resulting list of constituent sets of valid three taxon statements with the list of partial monothetic sets, now representing the collection of clada under option 4.

By generating three taxon statements we can, within practical limits, explore the situation where, according to Wilkinson (1991) parsimony analysis will not be misled if "... for any pair of sister taxa A and B there is more reliable evidence of their membership in a series of nested holophyletic groups to the exclusion of any unrelated taxon C, than there is misleading counterevidence for the inclusion of either A or B in an alternative set of nested groups to the exclusion of the other. "

Nelson and Platnick (1991) suggest another implementation of three taxon statements. They only consider all pairs of taxa, and disregard groupings of higher order, that can be derived from a list of taxa sharing the same state of a character. To form three taxon statements these pairs are united with all other taxa not sharing this state (i.e. having a zero), one at the time. This is repeated for each separate character. In this way a new data matrix is build, composed of the three taxon statements implied by the characters in the original data matrix. Note that my implementation of three taxon statements does not replace the original data matrix but only serves to provide additional building blocks for cladograms. Nelson and Platnick's definition of three taxon statements is also treated in this way by CAFCA. Only if you export the data matrix in NEXUS, PAUP, of HENNIG86 format the N&P three taxon codes will replace the original data matrix. If you opt for three-taxon-statements in the CAFCA parameter

dialog you will be offered a choice between Nelson and Platnick's implementation and CAFCA's.

PRIMARY ANALYSIS WITH PARTIAL MONOTHETIC SETS

In our first example the clada are derived by using option 1, partial monothetic sets (PMS), and are listed in table 3.3a. The numbers in the left margin of this table are the index numbers of the clada (cladon nrs). In the upper table the figures following each cladon number represent the index numbers of the terminal taxa (row numbers of the data matrix). E.g., cladon #9 is group {1 2 3} of terminal taxa.

All terminal taxa are included although they may lack a unique character state (e.g., taxon 3 and 4).

In the lower table the figures following each cladon represent the index number of the character states (column numbers of the binary data matrix) characterising the cladon. E.g., row #9 in this second list points out that group {1 2 3} (= row 9 in the first list) has column 15 from the binary data matrix as a unique character state.

Partial Monothetic Sets of terminal taxa in PLANT

1	1
2	2
3	3
4	4
5	5
6	3 4
7	4 5
8	3 4 5
9	1 2 3
10	2 3 4 5
11	1 2 3 4
12	1 2 3 4 5

Partial Monothetic Sets of character states in PLANT

1	2 5 13
2	6 8 11
3	
4	
5	10
6	9
7	14 16 20
8	7 12 19
9	15
10	3 4 18
11	17
12	1

Table 3.3a Building-blocks for cladograms (clada) with corresponding character states

Cladon numbers from these tables are also used in the apomorphy and state change lists, and as labels for the internal nodes of the diagrams. Note that **all** sets are used in the subsequent steps of the analysis, and not just the ones with the highest number of defining character states!

Using all the partial monothetic sets of terminal taxa from table 3.3a we can build a compatibility matrix (table 3.3b) in which the compatibility (in- or exclusion) for pairs of sets is indicated by a 1 entry. When we search this compatibility matrix with a branch and bound algorithm (Bron and Kerbosch, 1973) for the largest sets of mutual compatible clada (sets of terminal taxa) we find two maximal cliques, i.e., cladograms.

	1	2	3	4	5	6	7	8	9	10	11	12
1	1	1	1	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	1	1	1	1	1	1	1
6	1	1	1	1	1	1	0	1	0	1	1	1
7	1	1	1	1	1	0	1	1	1	1	0	1
8	1	1	1	1	1	1	1	1	1	1	0	1
9	1	1	1	1	1	0	1	1	1	0	1	1
10	1	1	1	1	1	1	1	1	1	0	1	0	1
11	1	1	1	1	1	1	0	0	1	0	1	1
12	1	1	1	1	1	1	1	1	1	1	1	1

Table 3.3b Compatibility matrix for sets of terminal taxa in table 3.3a.

CHARACTER STATES ON THE ROOT

The character states on the root of the selected cladogram(s) indicate the **polarity** (start) of each transformation series (table 3.4).

The **order** (sequence) of the transformation series can be read off from the cladogram, c.q., the list of state changes (table 3.6).

```
Character states on root for PLANT
----------------------------------
Rownumbers refer to index numbers of cladograms.
Columnnumbers refer to columns of multi-state data matrix.
     1  2  3  4  5  6  7  8  9 10
  +-----------------------------
2 |  1  1  1  0  0  2  0  1  2  0
```

Table 3.4 Character states on the root of selected cladograms.

Note that the root-state for characters 4, 5, and 10 is zero, as indicated in the CAFCA parameter dialog shown in the tutorial (fig. 3.11). These characters are multi-state characters, with zero included as a state. Zero's in multi-state characters (**not** in binary characters) are *forced* to be present on the root, as they are interpreted as putative ancestral states, *unless* you indicate otherwise in the CAFCA parameter dialog. This (default) interpretation of zero's in multi-state characters may influence the number of steps in a cladogram considerably. The inclusion of zero as a state in a multi-state character should therefore be considered rather carefully.

Cladogram optimisation.

Character states on the internal nodes of the cladogram are estimated by the optimisation algorithms described in Maddison and Maddison (1992, pp 92-96). Only the ACCTRAN option is implemented in CAFCA. I agree with dePinna (1991) in favouring ACCTRAN over DELTRAN because "... it better conforms with the notion that the conjecture of primary homology should be held valid unless demonstrated false by parsimony considerations." With the notion of primary homology dePinna refers to the stage of generating of a homology proposition, "... primary homology is a statement of putative generality, and expectation that correspondences are part of a general pattern." He speaks of secondary homology after the initial proposition has been tested by congruence and passed.

Another reason to prefer ACCTRAN is the protection it provides in estimating cladogram length in case of missing (i.e., fossil or otherwise unknown) taxa.

SELECTION CRITERIA

Eleven different selection criteria for cladograms are presented (table 3.5), six of which can be chosen in the CAFCA parameter dialog.

```
Selection criteria for cladograms of: PlantÆB
Column numbers refer to numbers of cladograms
---------------------------------------------
Row 1 : Total number of homoplasious events
Row 2 : Total number of single origins (Support)
Row 3 : Corrected Extra Length (x1000; CEL: Turner + Zandee)
Row 4 : Total number of state changes (S: Steps)
Row 5 : Redundancy Quotient (x1000; RQ: Zandee + Geesink)
Row 6 : Rescaled Redundancy Quotient (x1000; RQc)
Row 7 : Consistency Index (x1000; CI), with autapomorphy correction
Row 8 : Rescaled Consistency Index (x1000; RC: Farris)
Row 9 : Average Unit Character Consistency (x1000; AUCC: Sang)
```

```
Row 10: Homoplasy Distribution Ratio (x1000; HDR: Sang)
Row 11: Compatible Character State Index (x1000; CCSI: Zandee)

          1     2
    ------------
 1 |      2     0
 2 |     14    16
 3 |   3017     0
 4 |     18    15
 5 |    494   513
 6 |    146   178
 7 |    727  1000
 8 |    625  1000
 9 |    875  1000
10 |    250  1000
11 |    682   818

No-Order Limit for Steps, Extra Steps, RQ, and CI:

   S   ES   RQ   CI
-------------------
  27   12  408  400
```

Table 3.5 Selection criteria for cladograms of PLANT.

1. The first criterion considers the number of ad hoc statements (homoplasious events), i.e., all character states requiring more than 1 state change to explain their distribution over terminal taxa in the cladogram (reversal, parallelism, convergence). A single event of homoplasy contains at least 2 independent origins of a character state or the combination of at least 1 origin and 1 reversal.
Note that apomorphies may have contributed to such events as each non-single origin counts as a homoplasy and apomorphies may show multiple origins as long as they comply with the outgroup rule.

2. The total number of single origins. Only character states that require 1 state change for their origin to explain their distribution over taxa enter the second criterion. (In CAFCA versions previous to 1.5e the state was allowed to reverse to the condition on the root one or more times and still be counted as a state with a single origin).

3. CEL: the Corrected Extra Length (Turner and Zandee, ms), is calculated as

$$CEL = (\mathbf{G} - \mathbf{S}) + 1 - (\Sigma ri)/n$$

i.e., the number of extra steps in the cladogram as compared with the theoretical minimum, plus 1 minus the average unit retention index (**g - s)** / (**g- m**) (ri: Farris, 1989) for the n characters in the cladogram. CEL may be different for cladograms of the same length and can therefore offer an opportunity to select among most parsimonious cladograms (see also Rodrigo, 1992, and Sang, 1995).

4. The fourth criterion is a classical criterion of parsimony and is used as a default in CAFCA. It considers all steps needed to explain the distribution of all character states, without differentiation as to the quality of these steps.

5. The Redundancy Quotient (RQ; Zandee and Geesink, ms) minimises statements of homoplasy while maximising statements of homology at optimum levels of generality.

$$RQ = 1 - Hs / Hmax$$

$H_{max} = \log_2 A$

A is the product of *N* and *S*. *N* is the number of nodes in a completely resolved cladogram. *N* equals 2*T*-1, where *T* is the number of terminal taxa. *S* is the number of steps in a completely **un**resolved cladogram (bush).

$H_S = - \sum p.\log_2 p$

p = probability of character state change

$\sum p = 1$

These probabilities of character state change (or 'costs' of state changes) are determined by the total number of nodes in a cladogram, whether the nodes have supporting character states (versus zero branch length), the 'weight in nodes' that is carried by a node, as well as the 'weight in nodes' that supports a node, and, last but not least, whether the nodes branch off dichotomously or as polytomies. The contribution of a character is determined by the number of times its states change on the cladogram (steps), as well as by the frequency of character state changes (which can be zero !) on each of the branches of a cladogram.
Note that these probabilities are not (directly) based on assumptions regarding the *processes* involved in the evolution of the organisms studied. They only relate to implied properties of the cladogram, the character optimisation, and the data matrix used.

6. The Rescaled Redundancy Quotient (RQ_C).
RQ is rescaled to have a lower limit of zero by applying the formula:

$RQ_C = (RQ - RQ_{lower\ limit}) : (1-RQ_{lower\ limit})$

RQ_C is taken to be zero if RQ is equal to its lower limit.
$RQ_{lower\ limit}$ is de value of RQ on an unresolved cladogram (bush), given the data matrix.

7. The Consistency Index (CI) is the ratio of the theoretical minimum of steps given the number of character states (**M**), and the actual number of steps (**S**) needed in the cladogram to explain all distributions of character states over taxa (Kluge and Farris, 1969). In CAFCA aut-apomorphies do not enter the computation of the CI.

8. The Rescaled Consistency Index (RC: Farris, 1990), or, the product of the Consistency Index and the Retention Index (RI: Farris, 1989):

$RC = \{(\mathbf{G} - \mathbf{S}) / (\mathbf{G} - \mathbf{M})\} \times (\mathbf{M} / \mathbf{S})$

i.e., the ratio of the difference between the number of steps in a completely unresolved diagram (G) and the number of steps needed in the most parsimonious explanation of character state distributions on the actual diagram (S), and the difference between G and the minimum number of steps required to explain all character states as single origin events (M), times the ratio of M and S.

9. The average unit character consistency (AUCC; Sang, 1995), calculated as the average of total unit character consistencies:

$AUCC = \left[\sum c(i)\right] / n$

where c(i) is the unit character consistency (UCC) of character i (Kluge and Farris, 1969). AUCC is maximised when homoplasy is distributed most asymmetrically, i.e. all the homoplasy occurs in one character. AUCC actually varies in the interval [CI, 1] (Sang, 1995).

10. The homoplasy distribution ratio (HDR; Sang, 1995) is calculated as the ratio of the homoplasy distribution index (HDI; Sang, 1995) to the homoplasy index (HI; Sang, 1995)

HDR = HDI / HI

where

HDI = AUCC - CI, and

HI = 1 - CI

Since, whenever homoplasy occurs, the AUCC is smaller than 1, AUCC-CI is smaller than 1-CI. Thus, the HDR exists in the interval [0,1] (Sang, 1995). According to Sang (1995), HDR measures level of homoplasy and its distribution and can be a relatively accurate indicator of reliability of parsimonious cladograms. Although a cladogram may have a relatively low CI, it still can be considered reliable if its HDR is high, because in such a case the homoplasy is concentrated in a small group of cladistically unreliable characters.

11. The compatible character state index, **CCSI**, is calculated as the ratio of the number of compatible character *states*, i.e., the character *states* that are identical with components of the cladogram, and the total number of character *states*. Note that the CCSI does not measure compatibility among states or among characters directly. Autapomorphies are **not** excluded, although always consistent and thereby inflating the value of CCSI. This also applies to uninformative character states, i.e. states that are present in all the taxa concerned. For polarised multi-state characters the state that is assumed to be the start of the transformation series, and thus present at the root, is also considered uninformative.

CCSI varies in the interval [0,1]. It is zero for a bush, and reaches its maximum value when all character states are consistent with the cladogram. CCSI is a measure of “goodness of fit” for primary homologies.

CCSI is related to OCCI (Rodrigo, 1991). OCCI counts the number of fully compatible characters for a cladogram. OCCI is meant to be used as a discriminator for MPT’s.

CCSI also measures taxonomic efficiency in the sense of Rodrigo (1991); "the ease of which we may identify taxon membership in practice". Character states compatible with the cladogram have a single origin and as such represent unique identifiers for clades.

CCSI is related to Sharkey's (1989) proposal to use the degree of compatibility for characters as a means to choose among MPC's. For each character the degree of compatibility is measured by the expected number of incompatibilities and in proportion to this number character weights are derived. A standard parsimony method is used to find all MPC's. For each cladogram the list of character state changes is used to find those MPC's which call for the fewer changes in the more compatible characters.

Penny (1982a) showed for a data matrix comprising 20 genera of Epicridaceae with 18 binary characters that for a character "... the number of incompatibilities is a good guide to the number of duplications that will occur on the tree." Duplications are extra steps needed to explain the distribution of states over terminal taxa in a cladogram. Penny and Hendy

(1985) plotted the proportion of the maximum possible number of duplications against the ratio of observed and expected incompatibilties for hemoglobin b sequence data and showed a strong possitive correlation demonstrating that this ratio gives a good prediction of character reliability.

Starting from these observations for individual characters we would expect that their corresponding summary statistics like CI and CCSI would behave correspondingly. However, the relation between CI and CCSI is not monotonous and there is no linear correlation between number of steps for a cladogram and the number of character states compatible with it. Longer cladograms may have more compatible character states than a most parsimonious one (MPC), or they may have less, as I will show in the examples from the literature at the end of this chapter. This seems rather counterintuitive. The states of a character are mutually compatible by definition (when there is no polymorfism in terminal taxa), that is, they either include or exclude one another and they do not overlap. Why and how is it, then, that one state of a character may be compatible with a cladogram and the others not? The following example serves to explain this phenomenon.

We have two multistate characters, and a given cladogram. First we will consider the case when the characters are ordered (as indicated by their additive binary code).

```
     I        II   character
   1 2 3   1 2 3 4 character states
Aus 1 0 0   1 1 0 1    ---|---\
Bus 1 1 0   1 1 0 1    ---|   |---\
Eus 1 0 0   1 1 0 0    -------/   |---
Cus 1 1 1   1 0 1 0    ---|       |
Dus 1 1 1   1 0 1 0    ---|-------/
```

For two characters to be compatible they may show no conflict in the way they subdivide the set of terminal taxa in subsets according to their states. As these subsets for I and II contradict each other (e.g., BCD vs ABE) character I and II are not compatible in the ordered case. The OCCI score of compatible characters is one out of two. The unit CI (UCC) for character I is 0.667 and for character II equal to one. The ensemble CI for the cladogram is 0.8333 and the average unit CI (AUCC) is 0.8335. Looking at separate character *states* instead of characters we note 6 compatible states out of a maximum of 7 (CCSI = 0.8571).

```
     I        II   character
   1 2 3   1 2 3 4 character states
Aus 1 0 0   0 0 0 1    ---|---\
Bus 0 1 0   0 0 0 1    ---|   |---\
Eus 1 0 0   0 1 0 0    -------/   |---
Cus 0 0 1   1 0 0 0    ---|       |
Dus 0 0 1   0 0 1 0    ---|-------/
```

In case both characters are considered unordered (see above) the situation is different, although the characters I and II are still incompatible (contradiction in set membership of terminal taxa, e.g., AE vs AB). This may seem odd as their UCC's are now the same and equal to one. According to another definition of character compatibility "two characters are compatible if there is at least one hypothesis of evolutionary relationship (i.e. tree)

that is consistent with both characters." (Estabrook and Anderson, 1978). Both characters are consistent with the cladogram, but nevertheless they are in conflict as to set membership of terminal taxa.

The origin of this seemingly paradoxical situation is hidden in the way we consider state 1 for character I. Regarding set membership of terminal taxa this state points to the group {AE}. Optimisation of character states on the cladogram, however, assigns state 1 to the set {ABE} and maybe to the root {ABCDE} as well (as in the ordered case). Thus, before optimisation there is character state (= set membership) conflict and therefore incompatibility of characters. After optimisation the interpretation of state 1 in character I changes, and thus the conflict is resolved. Compatibility is an *a priori* predicate of a relation among characters or character states or sets of taxa, based on a primary homology assessment prior to cladogram estimation and optimisation. Consistency of a character or character states with respect to a cladogram as measured by the UCC is an *a posteriori* predicate of a relation between a character or character state and a cladogram (tree), after a congruence test that primary homologies may pass and become secondary homologies.

Returning to our example we observe that the unit CCSI for character I is still 0.667 as it was in the ordered case. The CCSI for the cladogram is 6 out of 7, also the same as in the ordered case. The CI for the cladogram, however, changed from 5 over 6 when characters are ordered to 1 in the unordered case, because a possible conflict was resolved after interpretation (optimisation). Taxon B also shows state 1 for character I but in its derived form, i.e., state 2. As a raw observation, however, the taxon set {AE} based on state 1 is still incompatible with the cladogram. **Cladogram optimisation does not change our compatibility assessments and thus the relation between CCSI and CI can not be monotonous**. Only when we revise our initial assessments (whether for characters or character states) as a result of cladogram optimisation analogous to the estimation of the number of steps for a character, the relation between CI and CCSI will be 1:1.

NO-ORDER LOWER LIMIT

The CI can not be lower than M/G. In practice, even a completely unordered cladogram, i.e., one big polytomy (bush), has a CI that does not equal zero; only RC is defined to be zero in that case.

As a last entry, the actual value for these parameters in the case of an unordered cladogram (bush) is given to serve as a point of reference, the *no-order lower limit*.

You can rescale CI (as is done in RQ_C) to have a lower limit of zero by applying the formula

$$CI_C = (CI - CI_{\text{lower limit}}) : (1-CI_{\text{lower limit}})$$

CI_C is taken to be zero if CI is equal to its lower limit.

UNINFORMATIVE CHARACTERS

Character 1 in table 3.2 is uninformative as it shows one state present in all taxa. This state counts as a supporting state, although its supposed single origin cannot be inferred from the cladogram and the data matrix as given. It is therefore not counted as a step on the cladogram (except in criterion 6), but it receives the benefit of the doubt as support.

DIAGRAMS, APOMORPHIES, COMPATIBILITIES, AND STATE CHANGES

The terminal nodes of the cladogram are labelled by the taxon numbers (= row numbers of the data matrix) and their names (if present). The internal nodes of the cladogram are labelled using the cladon numbers from the list of monothetic sets (clada). For instance, group {3 4 5} from the cladogram is entry # 8 from the list of clada (table 3.3).

As already explained states on internal nodes of a cladogram are estimated by algorithms as specified in Maddison and Maddison (1992). Only the ACCTRAN option is implemented in CAFCA.

When all nodes of the cladogram have their appropriate label we can easily enter the state changes for characters onto the branches of the cladogram by using the table accompanying the cladogram [table with state changes for each character with reference to cladon nrs]. State changes that can be considered evolutionary novelties if tested by an outgroup (local for internal nodes, global for the root) enter the list of apomorphies [with reference to cladon nrs].

Character states that are fully compatible with the cladogram are listed in the middle part of the table under the heading ‘compatibilities’. These compatibilities are not identical with steps on the cladogram (no single origins nor apomorphies), just congruencies between groups in the cladogram and character states. They can’t be steps as all states from a character may be involved, like for instance character 2 with state 1 on cladon 1 and state 2 on cladon 10. Both states of character 2 are congruent with a group in the cladogram but only one step is involved as can be seen in the list of state changes.

The relation between the notion of ‘character states compatible with the cladogram’ and the concepts of ‘*character* compatibility’ and ‘*group* compatibility’ is explained in the final section of this chapter.

Several character states (states on the root) enter the list of apomorphies although there is no global outgroup indicated to test their status; they receive the benefit of the doubt as evolutionary novelties.

```
Plant : Cladogram-2
                /--- 4 Dus
             /---7
      /---- 8   \--- 5 Eus
/--- 10    |
|     |    \------  3 Cus
|     |
|     \------------  2 Bus
|
\------------------ 1 Aus
```

PlantB: Cladogram–2 : APOMORPHIES

Cladon	Character	State
1	3	2
	6	3
2	6	1
3		
4		
5	5	3
7	7	1
	8	0
	9	3
	10	3
8	4	2
	5	2
	10	2
10	2	2
	4	1
	5	1

	10	1

PlantB: Cladogram–2: COMPATIBILITIES

Cladon	Character	State
1	2	1
	3	2
	6	3
2	4	1
	5	1
	6	1
5	5	3
7	7	1
	8	0
	9	1
	10	3
8	4	2
	6	2
	10	2
10	2	2
	3	1
	10	1
12	1	1

Plant : Cladogram-2: STATE CHANGES

Character	Cladon	Change

1		
2	10	1 -> 2
3	1	1 -> 2
4	8	1 -> 2
	10	0 -> 1
5	5	2 -> 3
	8	1 -> 2
	10	0 -> 1
6	1	2 -> 3
	2	2 -> 1
7	7	0 -> 1
8	7	1 -> 0
9	7	2 -> 1
10	7	2 -> 3
	8	1 -> 2
	10	0 -> 1

Characters and States refer to multi-state data matrix, if present.
Cladon nrs refer to the list of monothetic sets of terminal taxa.

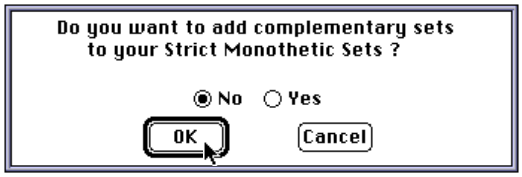
Table 3.6 Selected diagram with apomorphies, single origins, and state changes for first example.

This example shows how polarity decisions made a priori by you may change as a result of the analysis. If you decided *a priori* that the state observed for character 8 in taxa 1, 2, and 3 is the apomorphic one its polarity now appears to be reversed, or taxa 4 and 5 (cladon 7) show a reversal to an ancestral state (absence).

PRIMARY ANALYSIS WITH STRICT MONOTHETIC SETS

Our second example uses the same data matrix, PLANT, but now the clada used are strict monothetic sets (SMS) of terminal taxa, and the best cladogram is selected by the Redundancy Quotient (table 3.7). To run this example you can follow the same steps as given for the first example in the beginning of this chapter, except for the CAFCA parameters dialog, where you must now click the second button of the top rows instead of the first.

There is one other point where this example deviates from the first as to the dialogs and prompts you meet. After the CAFCA parameters dialog you must decide whether you will add complementary sets, as described under option 3 to your strict monothetic sets. Take the default **No** and click **OK**.



We now find 14 clada instead of 12 when using PMS (table 3.8). The group {2 3 4} is new, as well as {2 3}. They constitute alternatives for breaking down the group {2 3 4 5}, present in both PMS and SMS, in inclusive sets, due to the interaction of character state distribution 3 (or 4) and 18 with state distribution 17 and 15, respectively (table 3.2).

```
     CAFCA - Mac   version 1.3i  (c) M.Z. 1987-1995
          Date:1 AUG 1995     Time:9H35M28S
             CAFCA Parameter Settings
Type of analysis ..................... Primary
Cladon option ........................ 2: Strict  Monothetic Sets (SMS)
Cladogram Selection Criterion ......... Maximum Redundancy  Index
Taxon on outgroup node ...............
Ancestral zero's in character ......... 4 5 10
Ordered characters .................... 10
[Maximum] Number of Cladograms ........ 9
```

Table 3.7 Header of CAFCA output of strict monothetic sets example.

```
Strict Monothetic Sets of terminal taxa in Plant
------------------------------------------------
  1 |    1
  2 |    2
```

```
  3 |    3
  4 |    4
  5 |    5
  6 |    3 4
  7 |    4 5
  8 |    2 3
  9 |    3 4 5
 10 |    1 2 3
 11 |    2 3 4
 12 |    2 3 4 5
 13 |    1 2 3 4
 14 |    1 2 3 4 5
     ------------

Strict Monothetic Sets of character states in Plant
-------------------------------------------------
  1 |     1  2  5 13 15 17
  2 |     1  3  4  6  8 11 15 17 18
  3 |     1  3  4  7  9 12 15 17 18 19
  4 |     1  3  4  7  9 12 14 16 17 18 19 20
  5 |     1  3  4  7 10 12 14 16 18 19 20
  6 |     1  3  4  7  9 12 17 18 19
  7 |     1  3  4  7 12 14 16 18 19 20
  8 |     1  3  4 15 17 18
  9 |     1  3  4  7 12 18 19
 10 |     1 15 17
 11 |     1  3  4 17 18
 12 |     1  3  4 18
 13 |     1 17
 14 |     1
     --------------------------------------
```

Table 3.8 Clada with corresponding character states for strict monothetic sets example.

Using PMS (option 1), only {3 4} and {3 4 5} were available as subsets for {2 3 4 5}, due to character state 9 and 7 in the binary data matrix (table 3.2). The clada {2 3 4} and {2 3} have no unique separate character states but they do possess unique combinations of character states (table 3.8; # 8 and 11).

In general, the use of SMS as building-blocks for cladograms brings more resolving power in problems where the patterns implied by the distribution of character states over taxa do not allow for completely dichotomous cladograms.

In other cases, like the present example, the use of SMS simply adds alternatives to the set of already completely resolved cladograms, although not necessarily better ones. Nine alternative cladograms result from this analysis (table 3.9). One of them (# 6) is identical to the cladogram selected in the first example. Other criteria (e.g. number of homoplasious events) select cladogram # 5 as well.

```
Selection criteria for cladograms of: PlantÆB
Column numbers refer to numbers of cladograms
---------------------------------------------
Row 1 : Total number of homoplasous events
Row 2 : Total number of single origins (Support)
Row 3 : Corrected Extra Length (x1000; CEL: Turner + Zandee)
Row 4 : Total number of state changes (S: Steps)
Row 5 : Redundancy Quotient (x1000; RQ: Zandee + Geesink)
Row 6 : Rescaled Redundancy Quotient (x1000; RQc)
Row 7 : Consistency Index (x1000; CI), with autapomorphy correction
Row 8 : Rescaled Consistency Index (x1000; RC: Farris)
Row 9 : Average Unit Character Consistency (x1000; AUCC: Sang)
Row 10: Homoplasy Distribution Ratio (x1000; HDR: Sang)
Row 11: Compatible Character State Index (x1000; CCSI: Zandee)

        1     2     3     4     5     6     7     8     9
    ------------------------------------------------------
 1 |    2     3     5     3     0     0     2     5     6
 2 |   14    13    10    12    16    16    14    10     6
 3 | 3017  4033  6100  4083  1017     0  4033  7117  8183
 4 |   18    19    21    19    16    15    19    22    23
 5 |  492   460   449   473   487   510   471   432   437
 6 |  138    84    66   107   130   170   103    37    45
 7 |  727   667   571   667   889  1000   667   533   500
```

8	606	502	325	502	852	1000	502	248	178
9	875	860	817	902	975	1000	860	810	863
10	250	335	358	533	600	1000	335	401	605
11	682	545	455	636	682	818	500	409	500

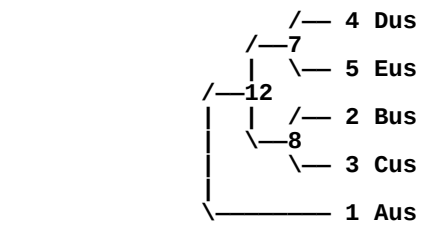
No-Order Limit for Steps, Extra Steps, RQ, and CI:

S	ES	RQ	CI
26	11	410	421

Table 3.9 Selection criteria for cladograms in strict monothetic setss example.

Comparing the state changes for characters in the best and second best cladogram from the second example (table 3.6 and 3.10) we see that a priori defined polarity and order in states as those of character 10 can be changed as a result of the analysis. Character 10 in cladogram #5 departs from the sequence as originally coded, 0 -> 1 -> 2 -> 3. There is now a state change 0 > 2 present, implying two steps as character 10 is ordered. This makes cladogram # 5 one step longer than # 6, though **not** due to a homoplasy.

Plant: Cladogram - 5



PlantB: Cladogram–5 : APOMORPHIES

Cladon	Character	State
1	3	2
	6	3
2	4	1
	5	1
	6	1
3		
4		
5	5	3
7	7	1
	8	0
	9	1
	10	3
8		
12	2	2
	4	2
	5	2
	10	2

PlantB:Cladogram–5: COMPATIBILITIES

Cladon	Character	State
1	2	1
	3	2
	6	3
2	4	1
	5	1
	6	1
5	5	3
7	7	1
	8	0
	9	3
	10	3
12	2	2
	3	1
	10	1
14	1	1

Plant: Cladogram–5 : STATE CHANGES

Character	Cladon	Change
1		
2	12	1 –> 2
3	1	1 –> 2
4	2	2 –> 1
	12	0 –> 2
5	2	2 –> 1
	5	2 –> 3
	12	0 –> 2
6	1	2 –> 3
	2	2 –> 1
7	7	0 –> 1
8	7	1 –> 0
9	7	2 –> 1
10	2	2 –> 1
	7	2 –> 3
	12	0 –> 2

Characters and States refer to multi–state data matrix, if present.
Cladon nrs refer to the list of monothetic sets of terminal taxa.

Table 3.10 Second best cladogram with apomorphies and state changes.

Note, however, that the sequence as estimated by CAFCA although not reflecting the original coding is fully consistent (= not implying homoplasious steps) with the different topology. We may tentatively conclude that although additive binary coding of character states defines only one particular character state tree, this coding may be consistent with more than one cladogram topology, as measured by the consistency index. This is due to the insensitivity of

this index to monitor the unambiguous reconstruction of (predefined) polarity and order in multi-state characters from cladograms.

PRIMARY ANALYSIS WITH ALL ADDITIVE BINARY CODINGS

Our third example again uses the same data matrix, PLANT, but now the clada used are the partial monothetic sets (PMS) of terminal taxa plus all additive binary codings possible for each block of homologous states (PMS + ABC). The best cladograms are selected by using the minimum step criterion. To run this example you can follow the same steps as given for the first example in the beginning of this chapter, except for the CAFCA parameters dialog where you must click the new settings, i.e., # 3: PMS + ABC.

We now find 16 clada instead of 12 when using PMS or 14 when using SMS (table 3.11). Compared with the first example the following groups are added to the list of clada: {1 2}, {2 5}, {2 3 4}, and {1 3 4 5}. It is clear from table 3.11 with sets of character states that these new clada do not possess at least one unique character state (clada 8, 9, 12, and 15).

Partial Monothetic Sets of terminal taxa in Plant		Partial Monothetic Sets of character states in Plant	
1	1	1	2 5 13
2	2	2	6 8 11
3	3	3	
4	4	4	
5	5	5	10
6	3 4	6	9
7	4 5	7	14 16 20
8	2 5	8	
9	1 2	9	
10	3 4 5	10	7 12 19
11	1 2 3	11	15
12	2 3 4	12	
13	2 3 4 5	13	3 4 18
14	1 2 3 4	14	17
15	1 3 4 5	15	
16	1 2 3 4 5	16	1

Table 3.11 Clada with corresponding character states obtained with option 4, PMS plus all additive binary codings.

These groups are based on an additive combination of character states within a block of homologous states. For instance, {1 2} is based on columns 11 + 13 [= state 1 plus 3 of character 6], {1 3 4 5} on columns 12 + 13 [= state 2 plus 3 of character 6], {2 5} on columns 8 + 10 [= state 1 plus 3 of character 5], {2 3 4} on 8 + 9 [= state 1 plus 2 of character 5].

In fact all additive codings possible within each block of homologous states are used to add new clada to the list that can be build on the basis of PMS only. (CAFCA poses a limit of 6 states per block, as a default, to derive additive codings, thus implying a maximum of 945 character state trees per character. You can change this default in the CAFCA parameter dialog, up to maximum value of 12. Note, however, that the number of implied cladograms increases exponentially relative to the number of taxa, such that 12 in this context is an extremely high number).

The use of all possible additive codings for each character adds considerable resolving power to the set of clada. It boils down to exploring and comparing all polarity and order decisions possible for each multi-state character (transformation series).

By means of the **Strict Sets** option in the **Print** menu we can generate the strict monothetic sets of character states, instead of only the partial sets, that characterise the clada resulting from the PMS+ABC option listed in table 3.11. These strict sets of character states indicate that some of the clada, as # 9 & 11, 8 & 13 and 15 & 16, besides lacking unique character states, don't even have unique *combinations* of character states that could be used to characterise them (table 3.12).

```
Strict Monothetic Sets of character states in Plant
---------------------------------------------------
  1|    1  2  5 13 15 17
  2|    1  3  4  6  8 11 15 17 18
  3|    1  3  4  7  9 12 15 17 18 19
  4|    1  3  4  7  9 12 14 16 17 18 19 20
  5|    1  3  4  7 10 12 14 16 18 19 20
  6|    1  3  4  7  9 12 17 18 19
  7|    1  3  4  7 12 14 16 18 19 20
  8|    1  3  4 18
  9|    1 15 17
 10|    1  3  4  7 12 18 19
 11|    1 15 17
 12|    1  3  4 17 18
 13|    1  3  4 18
 14|    1 17
 15|    1
 16|    1
     -------------------------------------
```

Table 3.12 Strict monothetic sets of character states for the clada based on PMS + all additive binary codings for multi-state characters.

In contrast to SMS (second example) your homology hypotheses are not broken down relative to the pattern exhibited by other states of other characters, but these hypotheses are merged (added) in all different permutations possible within each block of homologous states. These extended combinatorial possibilities produced by the increased number of mutually inclusive groups of terminal taxa result in a larger number of cladograms (12) compared to PMS alone.

```
Selection criteria for cladograms of: PlantÆB
Column numbers refer to numbers of cladograms
---------------------------------------------
Row 1 : Total number of homoplasous events
Row 2 : Total number of single origins (Support)
Row 3 : Corrected Extra Length (x1000; CEL: Turner + Zandee)
Row 4 : Total number of state changes (S: Steps)
Row 5 : Redundancy Quotient (x1000; RQ: Zandee + Geesink)
Row 6 : Rescaled Redundancy Quotient (x1000; RQc)
Row 7 : Consistency Index (x1000; CI), with autapomorphy correction
Row 8 : Rescaled Consistency Index (x1000; RC: Farris)
Row 9 : Average Unit Character Consistency (x1000; AUCC: Sang)
Row 10: Homoplasy Distribution Ratio (x1000; HDR: Sang)
Row 11: Compatible Character State Index (x1000; CCSI: Zandee)

         1     2     3     4     5     6     7     8     9    10    11    12
    ------------------------------------------------------------------------
 1 |     3     2     6     3     5     2     2     0     4     0     0     6
 2 |    13    14     8    13    10    14    14    16     9    16    16     6
 3 |  4033  4033  7117  4033  6100  3017  4033  1017  5100     0  1017  8183
 4 |    19    19    22    19    21    18    19    16    20    15    16    23
 5 |   466   475   435   460   449   492   460   488   459   510   472   435
 6 |    95   110    43    84    66   138    84   131    83   170   105    42
 7 |   667   667   533   667   571   727   667   889   615  1000   889   500
 8 |   502   502   248   502   325   606   502   852   409  1000   852   178
 9 |   860   860   810   860   817   875   860   975   892  1000   975   863
10 |   335   335   401   335   358   250   335   600   567  1000   600   605
11 |   545   545   455   545   455   682   545   682   636   818   682   500

No-Order Limit for Steps, Extra Steps, RQ, and CI:

   S   ES   RQ   CI
------------------
  26   11  410  421
```

Table 3.13 Selection criteria for cladograms in PMS+ABC example.

Using minimum steps as a selection criterion (table 3.13), only one cladogram sticks out as the best: nr 10. Cladogram # 10 is identical to the cladogram selected in our previous runs (table 3.6).

	1	2	3	4	5	6	7	8	9	10	11	12
1	4	3	4	4	5	2	2	0	1	0	0	1
2	11	11	11	11	9	14	14	16	14	16	16	14
3	4100	3050	3050	4100	4100	3050	3050	0	0	0	0	0
4	19	18	18	19	19	18	18	15	15	15	15	15
5	436	451	446	436	423	469	467	468	474	481	481	479
6	6	34	25	7	0	65	62	63	75	86	86	83
7	667	727	727	667	667	727	727	1000	1000	1000	1000	1000
8	0	208	208	0	0	208	208	1000	1000	1000	1000	1000
9	860	875	875	860	860	875	875	1000	1000	1000	1000	1000
10	335	250	250	335	335	250	250	1000	1000	1000	1000	1000
11	545	545	455	545	455	682	545	682	636	818	682	500

No-Order Limit for Steps, Extra Steps, RQ, and CI:

S	ES	RQ	CI
19	4	432	667

Table 3.14: Cladogram evaluation data for 12 cladograms resulting from a primary analysis using option 3 (PMS + ABC) and zero's in multi-state characters as non-ancestral state.

If, however, we run this analysis again but now with the zero's in multi-state characters 4, 5, and 10 as *not* indicating putative ancestral states (see CAFCA parameter dialog), we arrive at the following evaluation data for the 12 cladograms involved (table 3.14).

Not forcing the zero's in characters 4, 5, and 10 to be present on the root reduces the number of steps in almost all cladograms. Cladogram # 8, 9, 11 and # 12 now also count 15 steps, just like # 10 already did. As the number of steps on the bush decreases as well (26 vs 19) some of the cladograms found (# 1, 4, and 5) appear to be as bad as a bush as regards the number of steps.

Note the lower limit of the Redundancy Quotient (0.432) which, due to the reduction of the number of steps on a bush (20 versus 26), is higher than in the other runs (0.410).

A last item from table 3.14 that is well worth noting is that for MPT's with ci=1 the AUCC can not discriminate any further (its range is [ci,1]). However, the compatible character states index (CCSI, row 11) does discriminate among these MPT's and selects # 10 as the best. In this particular case CCSI is even a better performer than the RQ (RQ does not discriminate among # 10 and # 11).

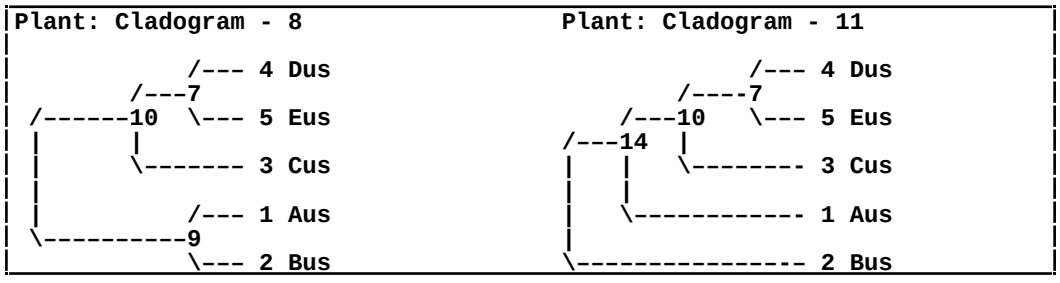


Figure 3.1: Alternative most parsimonious cladograms resulting from PMS+ABC and non-ancestral zero's in multi-state characters.

PRIMARY ANALYSIS WITH THREE-TAXON-STATEMENT PERMUTATIONS

Our fourth example again uses the same data matrix, PLANT, but now the clada used are the partial monothetic sets (PMS) of terminal taxa plus all clada resulting from valid three-taxon-statements (TTS's; table 3.15).

In comparison with the other examples run so far, many building blocks (30) for cladograms result from all possible three taxon statement permutations as, apparently, many of these statements are valid due to independent support for their constituent clada.

These building blocks give rise to 90 possible cladograms, only one of which is the shortest (16 steps; table 3.6) if we consider zero in a multi-state character to represent an ancestral state, forced to be present on the root. If we don't apply the latter option, three shortest cladograms (15 steps) result from the analysis, the same as from the third example (figure 3.1; PMS + ABC).

Partial Monothetic Sets of terminal taxa in Plant		Partial Monothetic Sets of character states in Plant	
1	1	1	2 5 13
2	2	2	6 8 11
3	3	3	
4	4	4	
5	5	5	10
6	3 4	6	9
7	4 5	7	14 16 20
8	3 5	8	
9	2 5	9	
10	2 4	10	
11	2 3	11	
12	1 4	12	
13	1 3	13	
14	1 2	14	
15	3 4 5	15	7 12 19
16	1 2 3	16	15
17	2 4 5	17	
18	2 3 5	18	
19	2 3 4	19	
20	1 3 4	20	
21	1 2 4	21	
22	1 4 5	22	
23	1 3 5	23	
24	1 2 5	24	
25	2 3 4 5	25	3 4 18
26	1 2 3 4	26	17
27	1 3 4 5	27	
28	1 2 4 5	28	
29	1 2 3 5	29	
30	1 2 3 4 5	30	1

Table 3.15 Clada, with corresponding character states, from valid three-taxon-statements.

As we can see from this table, many of the sets of taxa generated, do not have a character state by which they can be uniquely recognised. If we use the **Strict Sets** option in the **Print** menu to generate the unique *combinations* of character states for these sets of taxa, we see that many of them even lack these (table 3.16). For instance the sets # 9 {= 2 5}, 17 {= 2 4 5}, 18 {= 2 3 5}, and 25 {= 2 3 4 5} are all characterised by the combination of character states {1 3 4 18}, corresponding to characters 1^1, 2^2, 3^1, and 10^1.

Strict Monothetic Sets of character states in Plant	
1	1 2 5 13 15 17
2	1 3 4 6 8 11 15 17 18
3	1 3 4 7 9 12 15 17 18 19
4	1 3 4 7 9 12 14 16 17 18 19 20
5	1 3 4 7 10 12 14 16 18 19 20
6	1 3 4 7 9 12 17 18 19

```
  7|    1  3  4  7 12 14 16 18 19 20
  8|    1  3  4  7 12 18 19
  9|    1  3  4 18
 10|    1  3  4 17 18
 11|    1  3  4 15 17 18
 12|    1 17
 13|    1 15 17
 14|    1 15 17
 15|    1  3  4  7 12 18 19
 16|    1 15 17
 17|    1  3  4 18
 18|    1  3  4 18
 19|    1  3  4 17 18
 20|    1 17
 21|    1 17
 22|    1
 23|    1
 24|    1
 25|    1  3  4 18
 26|    1 17
 27|    1
 28|    1
 29|    1
 30|    1
     -------------------------------------
```

Table 3.16 List of unique combinations of character states (= strict monothetic sets) for the sets of taxa listed in table 3.15.

You could, as an alternative, also use option 4 with three-taxon-statements according to Nelson and Platnick (1991). In that case the monothetic sets of terminal taxa (clada) generated from the data matrix will be supplemented with all clada derived from all three taxon statements implied by the original characters. Remember that in contrast to Nelson and Platnick's original implementation in CAFCA the original data matrix is not replaced by the TTS's, except when the data matrix is exported. Table 3.17 shows the version in NEXUS format as exported by CAFCA (**Utilities** menu, **export data matrix** option).

The data matrix with extensions for three taxon statements according to the Nelson and Platnick (1991) definition generates 27 clada which give rise to 54 cladograms, one of which is most parsimonious with 15 steps. It is identical to the one found by the PMS option alone. Note that CAFCA's definition of three-taxon-statements generates more clada (30 vs 27) and as a consequence more cladograms (90 vs 54) than the Nelson and Platnick definition. This is due to the fact that N&P only consider all pairs of terminal taxa to perform as units in TTS's, while CAFCA also uses units consisting of larger sets of terminal taxa.

```
#NEXUS
BEGIN DATA;
_DIMENSIONS NTAX=5 NCHAR=60;
_FORMAT missing=? symbols="0~9";
__MATRIX
      Aus 0??0??0??0??0?0?0?0?0?0?11??110?0?0?000000000001?1??1000000
      Bus ?0??0??0??0??0?0?0?0?0?01111???0?0?01?1??11?1??111??1?1?1??1
      Cus 111??0??0??011??1111??11??111111??1111??1?11??1??111??11??1?
      Dus 1111111111111111??1111??0?0?0?1111???111???111?????111?111??
      Eus ??01111111111??1111??1111?0?0?0??1111???111???111000000???111
;
end;
BEGIN ASSUMPTIONS;
_OPTIONS deftype=unord;
end;
```

Table 3.17. Data matrix for Nelson and Platnick three taxon statements in NEXUS format suitable to be run by PAUP

An analysis of the data from table 3.17 with PAUP generates 15 most parsimonious cladograms (60 steps) without an included all-zero ancestor and 105 including an ancestor. That means that all trees for the taxa involved are possible. The *replacement* of the original data with all three taxon statements accord-

ing to Nelson and Platnick's definition, in this case appears to render the data phylogenetically useless (without any information as to phylogenetic structure). Using the three-taxon statements as a source for clada (components) only, instead as a replacement for the original data, as is done in the CAFCA implementation of N&P's definition, appears to make more sense.

A COMPARISON WITH CHARACTER COMPATIBILITY ANALYSIS.

The concept of character compatibility originated from ideas by Wilson (1965), Camin and Sokal (1965), and Le Quesne (1969, 1972) and was further developed by Estabrook (1972), Estabrook et al. (1975), Meacham (1981) and others. Meacham and Estabrook (1985) provides the most recent overview. Felsenstein (1982) offers an introduction in a wider perspective.

The following introduction is according to Meacham (1981, 1984). Character compatibility analysis is a technique to reveal patterns of agreement and disagreement among characters in a data matrix. Characters serve as a basis for comparing the taxa in the data matrix. Characters can have two of more states. Taxa are considered alike with respect to a character if they share the same character state. In this way a character allows the recognition of discrete classes (sets) of taxa. Accordingly, a qualitative character is defined as a set of character states, which are mutually exclusive and exhaustive subsets of the collection of terminal taxa in the data matrix. Characters in the latter sense are considered compatible if and only if there is no conflict in membership. If there is a conflict among two characters in the way they subdivide the set of terminal taxa in subsets according to their states, then these characters are considered incompatible. To put it otherwise: "Two characters are compatible if there is at least one hypothesis of evolutionary relationship (i.e. tree) that is consistent with both characters." (Estabrook and Anderson, 1978).

Characters can be tested pairwise as to their compatibility. If a character has two states, 0 and 1, two characters are compatible with each other if three or fewer of the four possible combinations of their states 00, 01, 10, and 11 occur in the data matrix. A collection of characters are mutually compatible if and only if they are all pairwise compatible; the so-called Pairwise Compatibility Theorem. This theorem has been proven for directed (= polarised = with known ancestor) and undirected two-state characters, as well as for multi-state characters which are polarised and ordered, or ordered but unpolarized (states connected in an undirected network). The proof of this theorem for unordered + unpolarized multi-state characters is still open. (Note that according to the Estabrook and Anderson [1978] definition given above two unordered unpolarized multi-state characters are compatible if both have a CI=1 for a given cladogram).

Character compatibility analysis is aimed at finding the largest sets of mutually compatible characters (or maximal clique). These sets are used to build cladograms. For each maximal clique found, characters incompatible with it are excluded from consideration. For this reason character compatibility has been severely criticised, for instance by Farris and Kluge (1979) who state that "... deletion of a character from consideration, however, does nothing to indicate which points of similarity in a character are the result of homoplasy and which are not. Specific parallelisms or reversals are not detected by such methods."

Wilkinson (1994) showed for binary data that character compatibility analysis and parsimony analysis can only differ when there are more than five terminal taxa present in the data matrix. This is so because in standard parsi-

mony methods the number of steps that a character can contribute to the length of the cladogram, its threshold (Felsenstein, 1981) is its maximum number of steps, i.e. its steps on an unresolved cladogram or bush. The maximum number of steps of a binary character is equal to the number of taxa scored for the minority state. For five (or fewer) taxa the minority is either 1 or 2. Thus, the threshold for any character will not exceed 2. Under this condition only uniquely derived characters provide support for a cladogram, rendering standard parsimony equivalent to character compatibility analysis (Felsenstein, 1981, p 192).

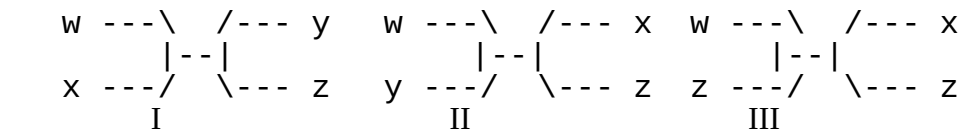
In exploring the relationship between character compatibility and group (or component) compatibility I will not use the same simple example data matrix with only 5 taxa as used in earlier paragraphs, but turn to examples from the literature, for the reason outlined above.

Penny (1982) showed the incompleteness of character (in)compatibility analyses in the sense that the (in)compatibility matrix used in the computations does not preserve and contain all the information present in the original data. In general it is not possible to get back to the original data given the (in)compatibility matrix. Different characterstate distributions over taxa may all lead to the same (in)compatibility matrix for characters. Besides the criticism against character compatibility outlined by e.g. Farris and Kluge (1979) the objections as raised by Penny (1982) are rather fundamental and analogous to those raised against distance data and phenetic approaches. I will show that these objections do not hold for group (or component) compatibility. Penny (1982) used the following character compatibility matrix in his example:

```
1 1 0
1 1 0
0 0 1
```

and showed that the next 6 different data sets *a-f* for 4 taxa will all lead to this character compatibility matrix.

	a	*b*	*c*	*d*	*e*	*f*
w	AAA	AAA	AAA	AAA	AAA	AAA
x	AAB	AAB	BBA	BBB	BBA	BBB
y	BBA	BBB	AAB	AAB	BBB	BBA
z	BBB	BBA	BBB	BBA	AAB	AAB
	I	I	II	II	III	III



The numerals I, II, and III underneath the data set indicate which of the three possible cladograms for four taxa is minimal for the particular data set. It is clear that what a compatibility matrix should at least preserve is the information regarding the different cladograms, if it can not recover the information as to the data set itself.

In CAFCA, compatibility matrices are drawn for sets of terminal taxa and not for characters. If we consider all possible sets, excluding the empty set, for the 4 taxa w, x, y, and z in Penny's example we arrive at 15 different groupings: {w}, {x}, {y}, {z}, {wx}, {wy}, {wz}, {xy}, {xz}, {wxy}, {wxz}, {wyz}, {xyz}, and {wxyz}. When we draw a square compatibility matrix for these sets and put a 1 for each pair of sets that is compatible, i.e., either in- or excludes each other but does not overlap, it is easy to see that the 6 different data sets

given above, will render 3 different sets of terminal taxa and therefore 3 different compatibility matrices for sets of terminal taxa (i.e., *a* and *b* are the same , as are *c* - *d*, and *e* - *f*), whether we use PMS or SMS. CAFCA will recover cladogram type I as the MPC for data set *a* and *b*, cladogram type II as the MPC for data set *c* and *d*, and cladogram type III for data set *e* and *f*, indicating the preservation of phylogenetic (or at least cladogenetic) relevant information in its group compatibility matrix.

As I have shown earlier when explaining the CCSI as an optimality criterion, cladogram length as the sum of steps in separate characters on the particular cladogram is an *a posteriori* quality assessment, as are CI, RC, and RQ for that matter, only possible after cladogram optimisation. As a matter of fact, cladogram optimisation, at least the downpass in the algorithm, is **the method** to estimate the number of steps a character takes on a cladogram. Compatibility, on the other hand, either of characters, character states, or sets of terminal taxa, is an *a priori* assessment, i.e., prior to the computation of cladograms and optimisation of characters, based on primary homology assumptions.

From this point of view the contrast between parsimony methods and compatibility methods as cladogram finding tools is unbalanced. If compatibility methods *could know* what standard parsimony methods *need to know*, that is, an estimate of the character states on the inner nodes of the cladogram, assessment of compatibility, whether of characters, character states, or set membership of taxa, would benefit. Parsimony finds its base in tests of congruence among characters and character states, and the quality of the results of the analysis (cladograms) is assessed after this test. As a corollary, primary homologies have changed to secondary ones, now each on its proper level of generality. As a result characters and character states incompatible before the analysis may be compatible after interpretation of primary homologies (optimisation). Thus, as a cladogram finding tool, compatibility based on primary homologies can never hope to find what may be the case for secondary homologies, unless all possible secondary homologies are in one way or another considered as data in the search for cladograms. Compatibility and parsimony methods would be on the same footing if, during computations, the compatibility assessments could change during and as a result of optimisation, just as the estimate of the number of steps may change as a result of different optimisations.

The problem for compatibility methods for character data is, then, how to incorporate effectively and efficiently all possible secondary homologies if it wants to find all MPC’s. For indirect data, as in historical biogeography and cases of co-evolution, the situation is different as we will see in chapter 6.

As the next examples from the literature may show in exploring the relationship between character compatibility and group compatibility, the use of strict monothetic sets, or all possible additive binary codings, or three taxon statement permutations, etc.., can serve as approximations to the consideration of all possible secondary homologies.

DE PINNA'S EXAMPLE

dePinna (1991) offers an example (table 3.18) for which standard parsimony and character compatibility result in different hypotheses of relationship, "... so that the different implications of the two methods are immediately obvious."

According to dePinna (1991, p. 382) parsimony analysis yields four equally parsimonious cladograms (found with the program Hennig86). The strict

consensus tree, shown in figure 3.2a, is identical with one of the four cladograms. This cladogram has 34 steps. If we apply character compatibility to the same data matrix we find a largest clique formed of characters 1-22 and obtain the cladogram shown in figure 3.2b. This cladogram is longer, with 42 steps.

```
                1     1     2     2  2
          5     0     5     0     5  7
Aus 11111 11111 00000 00000 10000 00
Bus 11111 11111 00000 00000 10111 11
Cus 11111 11111 00000 00000 00111 11
Dus 11111 11111 00000 00000 00000 00
Eus 00000 00000 11111 11111 01000 00
Fus 00000 00000 11111 11111 01111 11
Gus 00000 00000 11111 11111 00111 11
Hus 00000 00000 11111 11111 00000 00
```

Table 3.18 Data matrix from dePinna (1991, table 1) to exemplify the difference between parsimony and character compatibility.

When we use CAFCA to find cladograms from the data matrix in table 3.18 the PMS option finds one cladogram, identical to the one in figure 3.2b. When we apply the SMS option we find 4 cladograms, one of which is most parsimonious with 34 steps and identical with the cladogram in figure 3.2a. (As all PMS's are included in the SMS's, we also find the cladogram from figure 3.2a as one of the four).

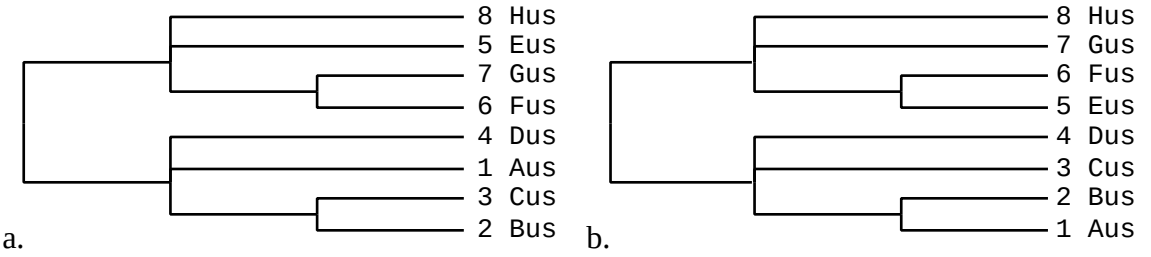


Figure 3.2 Cladograms for taxa Aus-Hus from data matrix in table 3.18 In (a) the arrangement based on parsimony ; in (b) the arrangement based on the largest clique of compatible characters (after dePinna, 1991).

Now it may seem at first sight as if group compatibility analysis based on partial monothetic sets of taxa is identical with character compatibility analysis. Second, if we extend the sets of terminal taxa considered by applying the definition for strict monothetic sets, and thus basing groups of taxa on partial agreement of characters, the results obtained by group compatibility analysis appears to be identical with strict parsimony analysis. To quote Farris and Kluge (1979), "Basing groups on partial agreement of characters not fully compatible seems more like a parsimony method than a clique procedure...".

Standard parsimony analysis makes clear that the primary homology assessments based on state 1 in characters 23-27, implying the set {BCFG}, are wrong. These states appear to be acquired two times independently. As secondary homologies there are in fact two different states 1 in these characters, one implies a taxon set {BC} and another implies {FG}. This is an *a posteriori* assessment. When these two different states 1 are considered from the point of view of character state compatibility they are compatible with the state 1 distribution in characters 1-22. We could conclude that *a posteriori* the largest clique of compatible characters now comprises 25 characters (vs 22 *a priori*).

FELSENSTEIN'S EXAMPLE

In considering character weighting and the use of likelihood methods for estimating phylogenies Felsenstein (1981) developed the threshold method as a

simple intermediate method between compatibility and parsimony for evaluating a phylogeny. He argued that "...We will frequently know that characters vary greatly in their evolutionary rates, but will not know which ones are the characters with high rates. In this case, weights can only be assigned to characters as part of the same process which produces an estimate of the phylogeny. Although this does not require us to know the weights of individual characters, if we know how variable the weights are this knowledge affects the process of estimating the phylogeny.' He continues to show for binary characters that when many characters evolve at low rates and a few may evolve quickly but we do not know in advance which characters will evolve at which rates the maximum likelihood estimate of the phylogeny produces the same result as character compatibility. In cases like this the contribution to the likelihood function made by the fraction of rapidly evolving characters is greater than that made by characters having two state changes, but less than that made by characters having just one state change. Considering T changes, instead of two, and its associated probability and its contribution to the likelihood function relative to the contribution made by the true fraction of rapidly evolving characters enables one to explore the middle ground between compatibility and standard parsimony methods as high values of T will produce maximum likelihood estimates that are identical to standard parsimony. In the threshold method we agree on a certain number for T and when evaluating a phylogeny we count T changes if a character requires more than T state changes, otherwise we simply count the number of changes.

Felsenstein (1981) offers the following example:

```
                1 11111 1
    12345 67890 12345 6
    ------------------
Aus 11111 11111 10011 1
Bus 11111 10000 00011 1
Cus 11111 10000 00000 0
Dus 00000 01111 10000 0
Eus 00000 01111 11111 0
Fus 00000 00000 01100 0
```

Table 3.19 Data from figure 2 in Felsenstein (1981)

For different values of the threshold T we obtain a different estimate of the cladogram for the taxa in the data matrix in table 3.19. When T=2 the character compatibility solution results (fig. 3.3a); for T=10 the standard parsimony solution results (fig. 3.3c), and for T=2.4 an compromise solution results (fig. 3.3b). Felsenstein (1981) depicts the cladograms as rooted phylogenies but acknowledges that the position of the root cannot be estimated in this case. For character weights equal to 1 these cladograms have a length of 28, 25, and 26 steps, respectively. As in dePinna's example it is clear that maximising compatibility among characters does not by necessity lead to cladograms of minimal length.

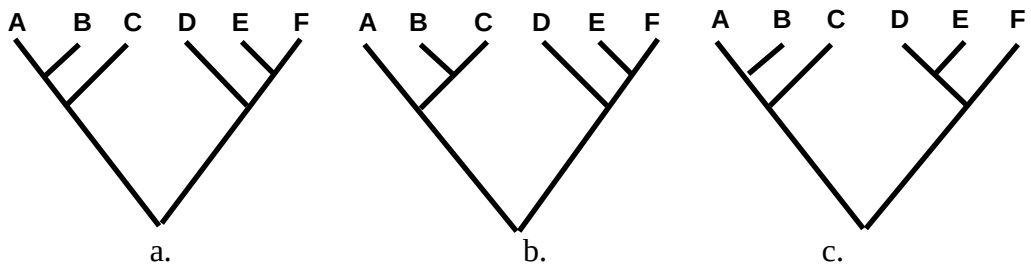
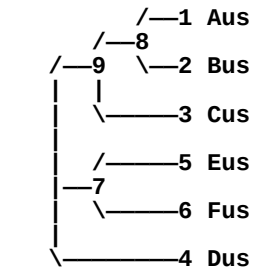


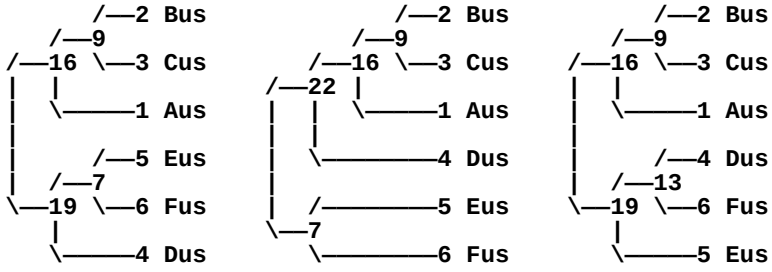
Figure 3.3 Estimates of the cladogram for the taxa in the data matrix in table 3.19 under three different values of the threshold T. (a) the character compatibility solution. (b) an intermediate solution (c) the standard parsimony solution.

When CAFCA is run with option PMS and zero's indicating ancestral states, 1 cladogram (shown below) results with 28 steps. It has a trichotomy at the root as the data do not indicate whether to unite {D} with {EF} or with {ABC}. The set {D} can be united with {EF} without extra costs in steps to obtain the character compatibility solution as depicted in fig 3a. If {D} is united with {ABC}, again without extra costs, a different rooting possibility for the same character compatibility solution results.

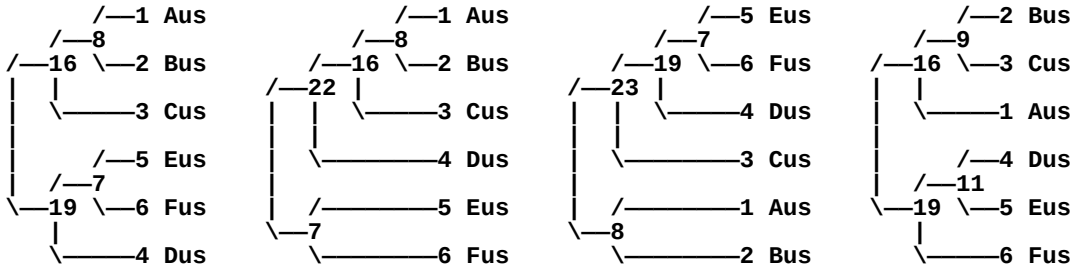


If we run CAFCA with the option PMS (no ancestral zero's) we get 3 cladograms with 28 steps, one of which is equal to figure 3.3a. Of the three solutions it has the highest CCSI (.938) and RQ (.455). The other two cladograms present different possibilities for the position of the root in figure 3.3a.

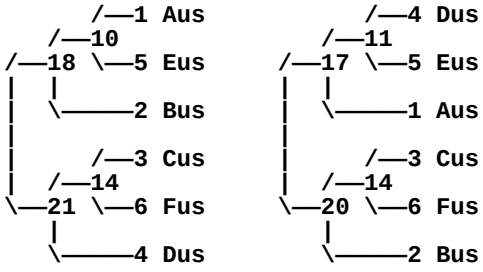
When CAFCA is run with option SMS (no ancestral zero's, no complementary sets) we find 30 cladograms, two of which are MPC's with 25 steps. One is identical with the standard parsimony solution depicted in figure 3.3c, the other shows a different possibility for the position of the root. Three cladograms count 26 steps (see below). The first one is identical to the intermediate treshold result (figure 3.3b). The second shows a different possibility for the position of the root. The third one is a new variety.



Four out of the 30 cladograms count 28 steps. They are all variants of the character compatibility result (see below).



Two other cladograms count 27 steps and can also be considered intermediate in this respect (see below). They probably have a threshold slightly higher than T=2.4



When we add the complementary sets to the option SMS (no ancestral zero's) we find 198 cladograms, 9 of which are MPC's with 25 steps. They correspond to the 9 rooted cladograms found with a standard parsimony approach (PAUP). It appears that in using this option all possible secondary homologies are incorporated in the sets of terminal taxa and as a consequence all MPC's can be found.

For 6 terminal taxa there are 945 possible rooted cladograms with labelled terminal nodes. When we evaluate all these 945 possibilities as user trees for the data matrix in table 3.19 we get the following results as to the behaviour of the different optimality criteria (figure 3.4).

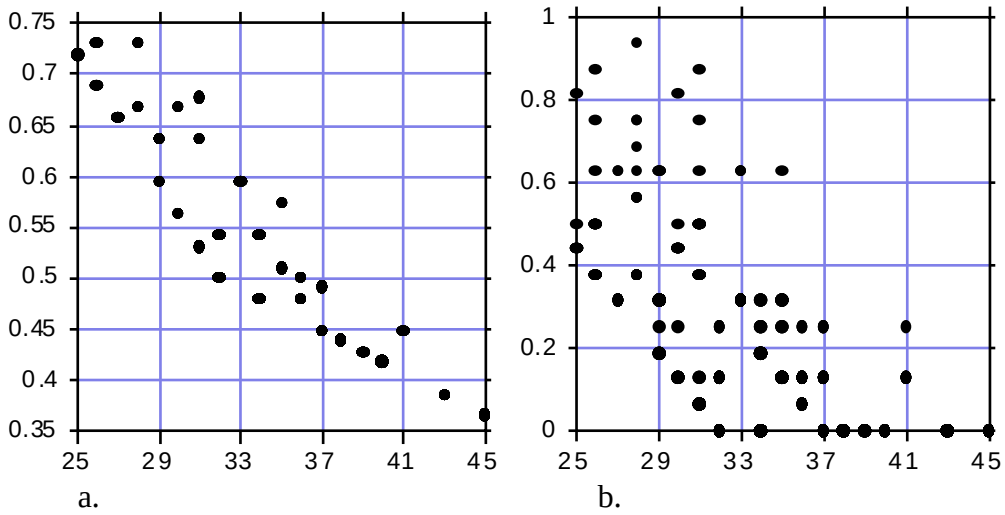


Figure 3.4 Relation between number of steps (x-axis) and AUCC (a; y-axis) and CCSI (b; y-axis) for all 945 cladograms for the six taxa in table 3.19

This particular example clearly shows that, at least for binary data, MPC's may neither exhibit a maximum value for the average unit character consistency (AUCC) nor for the compatible character state index (CCSI). As already stated by Felsenstein (1981) "...There are many ways in which we could have designed a criterion intermediate between parsimony and compatibility approaches." AUCC and CCSI each appear to represent such a criterion.

MEACHAM'S DATA

The examples by dePinna (1991) and Felsenstein (1981) are artificial in the sense that they are made for the occasion in order to show the difference between standard parsimony and character compatibility with relevance for the congruence test of homology. Let's see how this difference and the comparison with group compatibility works out with real data and take the data matrix from Meacham (1981, table 1; here table 3.20) as a third example.

All characters in this matrix are two-state characters, so as for the Pairwise Compatibility Theorem to hold, it does not matter whether we ignore the ancestor given in the matrix, or use it to polarise the characters.

		1	1	2
	char #	5	0	5	0
Viridis		12111	11111	11121	11111
Alba		12111	21111	21121	11121
Pallida		12211	11111	11121	11121
Coccinea		12111	11111	12122	11212
Brunnea		12111	11111	22122	12211
Caerula		12112	11122	12211	21111
Rubra		12112	12211	11111	21111
Nigra		21121	21111	12211	11111
Ancestor		11111	11111	11111	11111

Table 3.20 Data matrix from Meacham (1981, table 1)

Analysing this matrix with CAFCA under option 1 (PMS), characters unordered, and with taxon # 9 as outgroup to polarise the characters, yields the following results. There are 33 partial monothetic sets of taxa, including all terminal taxa, although they may lack a unique character state (e.g., taxon 1, 2, and 9). The sets of taxa correspond with the sets of character states given in table 3.21. All empty sets are removed from this list.

Partial Monothetic Sets of character states for Meacham ‘84 table 1.

Set #	Character State #
3 :	6
4 :	40
5 :	34
6 :	18 20
7:	14 16
8 :	2 3 8
10 :	10 32
11 :	12
12 :	22
13 :	26
14 :	30 36
15 :	38
16 :	27
17 :	23
18 :	24
19 :	28
20 :	9 31
21 :	11
22 :	21
23 :	25
24 :	29 35
25 :	37
26 :	1 4 7
27 :	5
28 :	13 15
29 :	17 19
30 :	33
31 :	39

Table 3.21 Partially monothetic sets based on Meacham’s table 1.

I expect this set of states to correspond with the set of compatible characters if all characters concerned are included with (at least) one of their two states (the complementary state is compatible by definition), and there is just one cladogram involved. In case more cladograms do result from the analysis, the list above should summarise at least half the states (for *this* data matrix) from all maximum cliques of characters.

CAFCA generates 30 cladograms on the basis of the 33 partially monothetic sets of terminal taxa. Only one cladogram is the most parsimonious one, with 25 steps (table 3.22).

I expect the characters fully congruent with this cladogram to comprise the set of compatible characters if group compatibility analysis (with PMS) might be considered identical with character compatibility analysis.

The list with character states compatible with the cladogram is given alongside in table 3.22. Empty clada are not listed.

All characters in the data matrix are (directed) two state characters. The states within a character are always compatible, unless one or more taxa are coded to be polymorphic. As the groups in the cladogram do not conflict as to membership of their constituent taxa, so don’t the characters that correspond uniquely with these groups. Meacham (1981) presents one maximum clique of 16 mutually compatible characters; char # 1, 3, 4, 7, 8, 9, 10, 17, 20, 15, 18, 5, 14, 16, 19, and 2. The lists are indeed identical.

```
Cladogram – 5

            /—— 4 Coccinea
         /——14
         |  \—— 5 Brunnea
         |
      /——19 /—— 2 Alba
      |  |——15
   /——26 |  \—— 3 Pallida
/——32 |  |
|  |  |  \————— 1 Viridis
|  |  |
|  |  |  /————— 6 Caerula
|  |  \——10
|  |     \————— 7 Rubra
|  |
|  \———————————— 8 Nigra
|
\——————————————— 9 Ancestor
```

Cladon	Character	State
3	3	2
4	20	2
5	17	2
6	9	2
	10	2
7	7	2
	8	2
8	1	2
	4	2
10	5	2
	16	2
14	15	2
	18	2
15	19	2
19	14	2
26	2	2

Table 3.22 Most parsimonious cladogram as found by CAFCA based on sets in table 3.21, and its list of compatible character states.

There is another statistic on which we can base the same conclusion, and that is the consistency index for characters. Characters that are *fully* compatible with a cladogram will have a CI equal to one. The list with CI's for characters for all 30 cladograms is given below. Cladogram # 5 has 16 characters with a CI=1. They are the same as in Meacham's list of compatible characters. The other cladograms, which show more steps, have fewer characters that are *fully* compatible with them.

Consistency Indices for Characters of Wagner80B

Columnnumbers refer to index numbers of characters
Rownumbers refer to index numbers of cladograms
Row 0 refers to mean c.i. over all cladograms

	1	2	3	4	5	6	7	8	9	10	11	12
0	1.00	.55	1.00	1.00	.63	.50	1.00	1.00	1.00	1.00	.50	.48
1	1.00	.50	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.33
2	1.00	.50	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.33
3	1.00	.50	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.33
4	1.00	.50	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.33
5	1.00	1.00	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.33
6	1.00	.50	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.25
7	1.00	.50	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.25
8	1.00	.50	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00	.50	.33
9	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
10	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
11	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
12	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	1.00
13	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
14	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
15	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	1.00
16	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	1.00
17	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
18	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
19	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
20	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	1.00
21	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.33
22	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
23	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.33
24	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
25	1.00	1.00	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
26	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.50
27	1.00	1.00	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.33
28	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.33
29	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.33
30	1.00	.50	1.00	1.00	.50	.50	1.00	1.00	1.00	1.00	.50	.33

	13	14	15	16	17	18	19	20
0	.77	.63	.90	.63	1.00	.90	.95	1.00

1	.50	1.00	1.00	1.00	1.00	1.00	1.00
2	.50	.50	1.00	1.00	1.00	1.00	1.00
3	.50	.50	1.00	1.00	1.00	1.00	1.00
4	.50	1.00	1.00	1.00	1.00	1.00	1.00
5	.50	1.00	1.00	1.00	1.00	1.00	1.00
6	.50	.50	.50	1.00	1.00	.50	1.00	1.00
7	.50	.50	.50	1.00	1.00	.50	1.00	1.00
8	.50	.50	1.00	1.00	1.00	1.00	.50	1.00
9	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00
10	1.00	.50	1.00	.50	1.00	1.00	1.00	1.00
11	1.00	.50	1.00	.50	1.00	1.00	1.00	1.00
12	1.00	.33	1.00	.50	1.00	1.00	1.00	1.00
13	1.00	.50	1.00	.50	1.00	1.00	1.00	1.00
14	1.00	.33	1.00	.50	1.00	1.00	1.00	1.00
15	1.00	.33	1.00	.50	1.00	1.00	1.00	1.00
16	1.00	.50	1.00	.50	1.00	1.00	1.00	1.00
17	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00
18	1.00	1.00	1.00	.50	1.00	1.00	1.00	1.00
19	1.00	.50	1.00	.50	1.00	1.00	.50	1.00
20	1.00	.33	1.00	.50	1.00	1.00	.50	1.00
21	1.00	.50	.50	.50	1.00	.50	1.00	1.00
22	1.00	.33	.50	.50	1.00	.50	1.00	1.00
23	1.00	.50	.50	.50	1.00	.50	1.00	1.00
24	1.00	.33	.50	.50	1.00	.50	1.00	1.00
25	.50	.50	1.00	.50	1.00	1.00	1.00	1.00
26	.50	.50	1.00	.50	1.00	1.00	1.00	1.00
27	.50	1.00	1.00	.50	1.00	1.00	1.00	1.00
28	.50	1.00	1.00	.50	1.00	1.00	1.00	1.00
29	.50	1.00	1.00	.50	1.00	1.00	1.00	1.00
30	.50	1.00	1.00	.50	1.00	1.00	1.00	1.00

Table 3.23 Consistency indices for characters in 30 cladograms generated by CAFCA, using PMS, from Meacham's table 1.

The cladogram in table 3.22 is not completely resolved; cladon 19 is a trichotomy. In chapter 4 we will see how to resolve such nodes by means of a secondary analysis. An heuristic search with PAUP results in two most parsimonious cladograms with 25 steps. One is identical with cladogram #5 given above. The other resolves the trichotomy and has also 16 fully compatible characters.

Table 3.24 lists both the number of steps for the cladograms found by CAFCA as well as the number of character states compatible with these cladograms. It appears that cladograms of the same length may show a different number of compatible *states*. Nevertheless there is a tendency for shorter cladograms to have more compatible states. Whether this tendency is a general feature remains to be seen when we take a look at more complex multi state data.

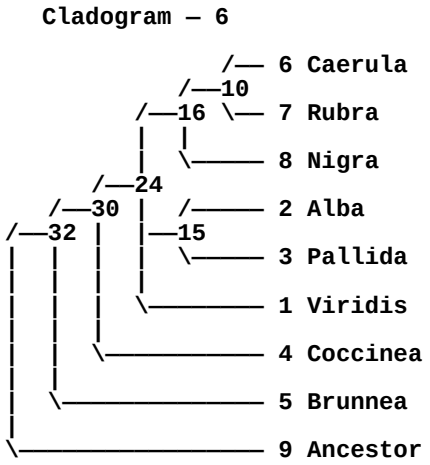
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
26	27	27	26	25	30	30	28	26	27	27	27	27	28	27	26	26	26	28	28	30	30	30	30	27	28	27	28	28	28
15	14	14	15	16	12	12	13	14	13	13	14	13	13	14	14	14	14	12	13	11	11	11	11	13	12	14	13	13	13

Table 3.24 Number of steps (row 2) and number of compatible states (row 3) for the 30 cladograms found by CAFCA for option PMS and undirected characters.

Cladogram #6 counts 30 steps. Its compatible character states are given in table 3.25. Empty clada are not listed. There are indeed fewer compatibilities left for this longer cladogram, 12 to be precise, instead of 16 for the shortest cladogram (you may also count the characters with CI=1 for cladogram # 6 in table 3.23). Most larger groupings in the cladogram, like cladon 24, 30, and 32, do not have a character state that is fully compatible with it.

I could tentatively suggest from this example that cladograms longer than the most parsimonious one, i.e., with more conflicts among character state distributions, have a set of compatible characters that is smaller than the maximum clique related to the shortest cladogram. However, we will also meet examples where the opposite holds, i.e., shorter cladograms have a smaller set of compat-

ible characters. In the introduction to these examples I have already discussed the probable cause of this phenomenon.

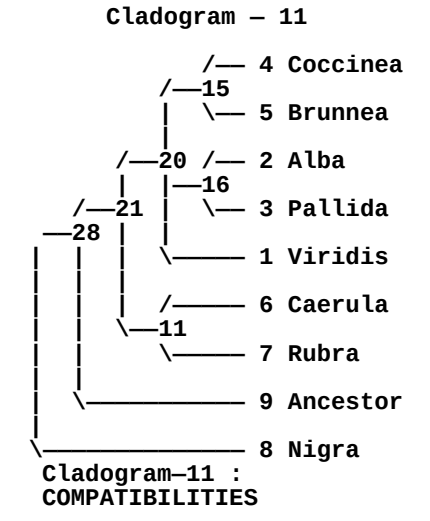


Cladogram–6 : COMPATIBILITIES

Cladon	Character	State
3	3	2
4	20	2
5	17	2
6	9	2
	10	2
7	7	2
	8	2
8	1	2
	4	2
10	5	2
	16	2
15	19	2

Table 3.25 One of the not-most-parsimonious cladograms and its compatibilities as found by CAFCA from Meacham's table 1.

What happens when we do not direct the characters by pinpointing an ancestor? CAFCA then finds 11 cladograms, all with 25 steps. Not one of these 11 cladograms is identical to the one given above, although # 11 comes close (table 3.26; it has taxon #9 'Ancestor' included in the in-group): Instead of 16 there are now 18 states listed, but two characters, # 1 and 4, are represented twice, due to the complementary clada 8 and 28. Thus the number of compatible characters is still 16, and the set is still identical with Meacham's (1981) maximum clique. As Meacham (1984) explains very clearly, directing characters can only increase conflicts, never remove them. It appears that in our first analyses, with taxon # 9 'Ancestor' as outgroup, no extra conflicts were introduced for cladogram # 5 by directing the characters. In general, however, adding an all-zero outgroup to a (binary) data matrix may very well turn compatible characters into incompatible ones by introducing the otherwise lacking fourth combination of character states (e.g., 0,0; or 1,1 in a multi-state matrix).



Cladon : Character : Character State		
2	3	2
3	20	2
4	17	2
5	9	2
	10	2
6	7	2
	8	2
7	1	2
	4	2
11	5	2
	16	2
15	15	2
	18	2
16	19	2
20	14	2
21	2	2
28	1	1
	4	1

Table 3.26 Cladogram and its compatible character states, found from Meacham's table 1 when characters are treated as undirected.

When we remove taxon # 9 'Ancestor' all together from the data matrix, CAFCA generates 30 cladograms from 31 partially monothetic sets, the same as

in our first analysis but now all 25 steps long. Cladogram # 5 (table 3.27) is identical to the already known topology, except for the absence of taxon # 9.

```
Cladogram – 5

          /—— 4 Coccinea
        /—13
        |  \—— 5 Brunnea
        |
      /—18 /—— 2 Alba
      |  |—14
   /—25  |  \—— 3 Pallida
   |  |  |
   |  |  \————— 1 Viridis
   |  |
   |  |  /————— 6 Caerula
   |  \—9
   |     \————— 7 Rubra
   |
   \———————————— 8 Nigra
```

Cladogram—5 : COMPATIBILITIES

Cladon	Character	State
2	3	2
3	20	2
4	17	
5	9	2
	10	2
6	7	2
	8	2
7	1	2
	2	1
	4	2
8	5	2
	16	2
13	15	2
	18	2
14	19	2
18	14	2
25	1	1
	2	2
	4	1

Table 3.27 Cladogram found by CAFCA from Meacham's table 1, while omitting the ancestral taxon.

Cladogram # 5 has 19 compatible character states, but some characters, like 1, 2, and 4, are represented twice, due to the fact that the complementary groups are both represented in the cladogram (clada 7 and 25). Thus the number of compatible characters is still 16, and it is the same set as Meacham's. In comparison, it is now made clear that introducing taxon # 9 'Ancestor' as an outgroup in our first analysis, and thus directing the characters, did raise the number of conflicts, or incompatibilities, among the characters for all cladograms other than # 5. In cladogram # 6, for instance, the number of conflicts raises with 5 as the number of steps increases from 25 to 30, and the number of characters compatible with the cladogram (and thus mutually compatible) decreases from 16 to 12.

```
Cladogram – 27

             /—— 4 Coccinea
           /—13
           |  \—— 5 Brunnea
           |
         /—18 /—— 2 Alba
       /—22 |—14
   /—25 |   |  \—— 3 Pallida
   |  |   |  |
   |  |   |  \————— 1 Viridis
   |  |   |
   |  |   \————————— 7 Rubra
   |  |
   |  \———————————— 6 Caerula
   |
   \——————————————— 8 Nigra
```

Figure 3.5 Cladogram with highest RQ, from analysis of Meacham's table 1 with ancestral taxon omitted.

Cladogram # 27 from this analysis has the highest RQ, but also the same 16 compatible characters. In this cladogram the group 'Caerula + Rubra' is broken down (figure 3.5). This cladogram also demonstrates that, if the most parsimonious cladogram, # 5, is not completely resolved, a longer cladogram, # 27, may show the highest RQ. However, after secondary analysis cladogram # 5 is completely dichotomous and still counts 25 steps but its RQc is now 0.123 in contrast to the secondary resolution of cladogram # 27 with a RQc = 0.116

The last option that CAFCA offers is running the analysis with all characters ordered and directed (i.e., transforming from state 1 to 2). This should not make any difference with the analysis with all characters directed through the introduction of an outgroup (having state 1 for all characters), because all characters have only two-states. Two-state characters can always be considered ordered (Meacham, 1984). Indeed, it does not make any difference. Only 1 cladogram is found from 18 partially monothetic sets, two of them empty (= with no supporting character state). It is the same as # 5 given above, with 25 steps, and 16 compatible characters.

STRAUCH'S DATA

Do these results imply that all *group* compatibility analyses under option 1 (Partial Monothetic Sets) will result in solutions identical with those of *character* compatibility analyses? So far, I only dealt with two-state characters. First, we need to compare the behaviour of CAFCA in dealing with multi-state characters, ordered en polarised as well as otherwise, before we can reach a definite conclusion.

I need another data matrix for that purpose, one with multi-state characters with more than two states. We may choose these data from other studies published on the application of character compatibility analysis, for instance the data on the *Alcidae* from Strauch (1984).

Data Matrix (multi–state) : Alcidae (Columns represent characters)

	1	2	3	4	5	6	7	8	9	10	11
1 Pinguinis_impennis	2	1	1	2	1	2	1	1	1	1	1
2 Alca_torda	2	1	1	2	1	1	1	1	1	1	1
3 Uria	2	1	1	2	1	1	1	1	1	1	1
4 Uria_aalge	2	1	1	2	1	1	1	1	1	1	1
5 Alle_alle	2	1	1	2	1	2	1	1	1	1	1
6 Cepphus_grylle	2	1	1	2	2	2	1	1	1	1	1
7 Cepphus_columba	2	1	1	2	2	1	1	1	1	1	1
8 Brachyramphus_marmoratus	2	1	1	2	2	2	1	1	1	1	1
9 Brachyramphus_brevirostris	2	1	1	2	2	2	1	1	1	1	1
10 Endomychura_hypoleucus	2	1	1	2	2	2	1	1	1	1	1
11 Synthliboramphus_antiquus	2	1	1	2	2	2	1	1	1	1	1
12 Synthliboramphus_wumizusume	2	1	1	2	2	2	1	1	1	1	1
13 Ptychoramphus_aleuticus	2	1	2	1	2	2	1	1	1	1	1
14 Cyclorhynchus_psittacula	2	1	2	1	2	2	1	1	2	1	1
15 Aethia_cristatella	2	2	2	1	2	2	1	1	2	1	2
16 Aethia_pusilla	2	2	2	1	2	2	1	1	2	1	2
17 Aethia_pygmaea	2	2	2	1	2	2	1	1	2	1	2
18 Cerorhinca_monocerata	1	1	1	1	2	2	2	2	1	2	2
19 Fratercula_arctica	1	1	1	1	2	1	2	2	1	2	2
20 Fratercula_corniculata	1	1	1	1	2	1	2	2	1	2	2
21 Lunda_cirrhata	1	1	1	1	2	1	2	2	1	2	2
22 OutGroup	1	1	1	1	1	1	1	1	1	1	1

	12	13	14	15	16	17	18	19	20	21	22
1 Pinguinis_impennis	2	1	2	2	1	2	2	3	1	1	1
2 Alca_torda	2	1	2	2	1	2	2	3	1	1	1
3 Uria	2	1	2	2	1	2	2	3	1	1	1
4 Uria_aalge	2	1	2	2	1	2	2	3	1	1	1
5 Alle_alle	1	2	1	2	1	2	1	2	1	1	2
6 Cepphus_grylle	1	1	2	2	1	2	1	2	1	1	2
7 Cepphus_columba	1	1	2	2	1	2	1	2	1	1	2
8 Brachyramphus_marmoratus	2	2	2	2	1	2	1	2	1	1	2
9 Brachyramphus_brevirostris	1	2	2	2	1	2	1	2	1	1	2
10 Endomychura_hypoleucus	2	1	2	2	1	2	1	2	2	1	2
11 Synthliboramphus_antiquus	2	1	2	2	1	2	1	2	2	1	2
12 Synthliboramphus_wumizusume	1	1	2	2	1	2	1	2	2	1	2
13 Ptychoramphus_aleuticus	2	2	1	1	2	2	1	2	1	1	2
14 Cyclorhynchus_psittacula	1	2	1	1	2	2	1	2	1	1	2

15	Aethia_cristatella	1	2	1	1	2	2	1	2	1	1	2
16	Aethia_pusilla	1	2	1	1	2	2	1	2	1	1	2
17	Aethia_pygmaea	1	2	1	1	2	2	1	2	1	1	2
18	Cerorhinca_monocerata	2	1	2	1	1	1	1	2	1	1	3
19	Fratercula_arctica	2	1	2	1	1	1	1	1	1	2	3
20	Fratercula_corniculata	2	1	2	1	1	1	1	1	1	2	3
21	Lunda_cirrhata	2	1	2	1	1	1	1	1	1	2	3
22	OutGroup	1	1	1	1	1	1	1	1	1	1	1

		23	24	25	26	27	28	29	30	31	32	33
1	Pinguinis_impennis	2	1	2	1	2	2	1	1	1	3	1
2	Alca_torda	2	1	2	1	1	2	1	1	1	3	1
3	Uria	2	1	2	1	1	1	1	1	1	3	1
4	Uria_aalge	2	1	2	1	1	1	1	1	1	3	1
5	Alle_alle	2	1	1	1	1	1	1	1	3	2	1
6	Cepphus_grylle	1	1	1	2	1	1	2	2	3	2	1
7	Cepphus_columba	1	1	1	2	2	1	2	2	3	2	1
8	Brachyramphus_marmoratus	1	1	2	2	2	1	2	1	3	3	2
9	Brachyramphus_brevirostris	1	1	2	2	2	1	2	1	3	3	2
10	Endomychura_hypoleucus	1	1	1	2	1	1	2	2	2	2	1
11	Synthliboramphus_antiquus	1	1	1	2	2	1	2	2	2	2	1
12	Synthliboramphus_wumizusume	1	1	1	2	2	1	2	2	2	2	1
13	Ptychoramphus_aleuticus	1	1	1	2	2	1	2	1	3	1	1
14	Cyclorhynchus_psittacula	1	1	1	2	2	1	2	1	3	2	1
15	Aethia_cristatella	1	1	1	2	2	1	2	1	3	2	1
16	Aethia_pusilla	1	1	1	2	2	1	2	1	3	2	1
17	Aethia_pygmaea	1	1	1	2	2	1	2	1	3	2	1
18	Cerorhinca_monocerata	1	1	1	2	3	1	2	1	3	1	1
19	Fratercula_arctica	1	2	1	2	3	1	2	1	3	1	1
20	Fratercula_corniculata	1	2	1	2	3	1	2	1	3	1	1
21	Lunda_cirrhata	1	1	1	2	3	1	2	1	3	1	1
22	OutGroup	1	1	1	1	1	1	1	1	1	1	1

Table 3.28 Data matirx for the Alcidae (Aves); after Strauch (1984), except the outgroup.

Strauch (1984) did not include an outgroup in his data matrix, but from the description he presents of the direction and ordering in the characters it is clear that an all-states-one outgroup is used. Most of the characters have only two states (and are thus always ordered). Only 5 characters have three states (# 19, 22, 27, 31, and 32). These latter characters constitute the real test case in our comparison of group- and character compatibility analysis. When ordered, multi-state characters are pairwise compatible if their binary factors are all pairwise compatible. Strauch's character compatibility analysis results in one maximum clique of 23 characters.

In order to make a direct comparison possible a CACFA primary analysis was run with option 1 (PMS), and all characters ordered, and directed by choosing taxon # 22 as outgroup. The analysis results in 50 partially monothetic sets for which 8 cladograms can be found. The table below presents the lengths of the cladograms as well as their number of compatible character states.

cladogram:	1	2	3	4	5	6	7	8
length:	67	70	68	71	72	75	73	76
compatible states:	55	56	54	55	51	52	50	51

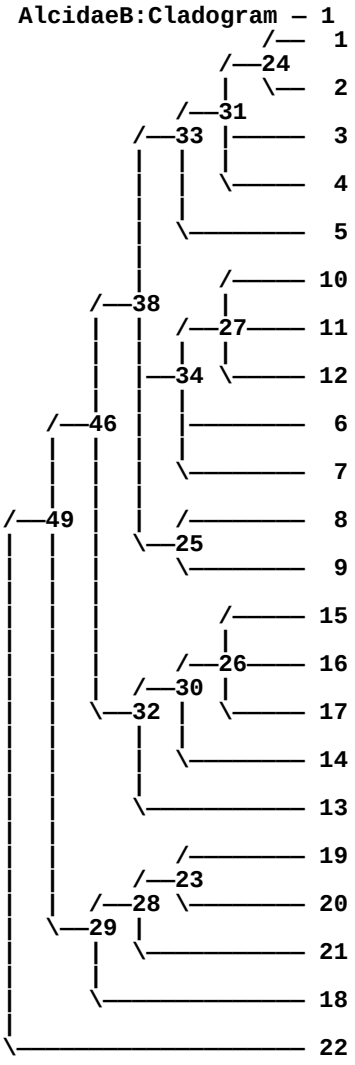
As was already noted with Meacham's data, longer cladograms tend to have less compatible character states. This relation is not monotonous as is indicated by cladograms # 5 and 8 which both have 51 compatible states but differ four steps in length. Cladogram # 2 is an exception to this tendency as it is three steps longer than # 1 but has 1 more state compatible with it.

Within the constraints of the search option there is only one most parsimonious cladogram with 67 steps. This cladogram as well as its list of compatible character states is given in table 3.29. Most striking in the list of compatibilities are the states for the root-node (# 50). In directing and ordering all characters, state 1 in the additive binary coding for the multi state characters is now present in all taxa, and for that reason also on the root of the cladogram.

Besides state 1 for all characters, there are 22 other states listed. Not all multi-state characters are present in this list; 19, 22 and 27 are there with states 1 and 3, characters 31 and 32 are only present with state 1 (only due to the process of ordering and directing the characters).

As all (binary expressions of the) states for multi-state characters must be pairwise compatible to make the characters compatible, the characters referred to above must be excluded as incompatible. All in all, the list of characters *fully* compatible with the cladogram (and therefore also mutually compatible) only counts 19 entries (in contrast to Strauch's 23). Some of the states of the other (incompatible) characters, however, are compatible with the cladogram, and CAFCA has used this information on *state*-compatibility in the search for cladograms.

When we compare the cladogram found by CAFCA with that given by Strauch (1984) we see that CAFCA's cladogram as presented in table 3.29 has all the groupings from Strauch's cladogram but is more resolved. As a consequence, less characters are fully compatible with it.

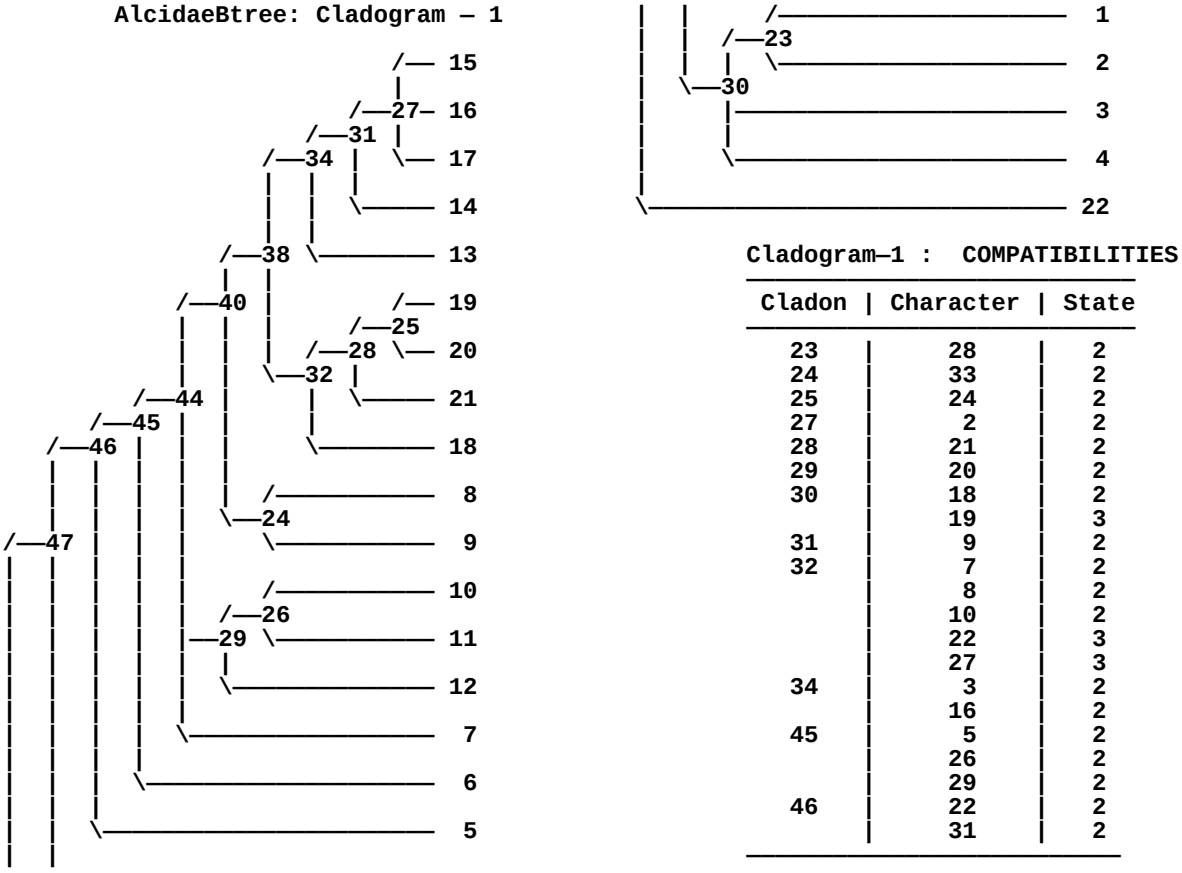


Cladogram–1 : COMPATIBILITIES

Cladon	Character	State
23	24	2
24	28	2
25	33	2
26	2	2
27	20	2
28	21	2
29	7	2
	8	2
	10	2
	22	3
	27	3
30	9	2
31	18	2
	19	3
32	3	2
	16	2
33	23	2
34	30	2
38	4	2
	15	2
46	1	2
	17	2
50	1	1
	2	1
	3	1
	4	1
	5	1
	6	1
	7	1
	8	1
	9	1
	10	1
	11	1
	12	1
	13	1
	14	1
	15	1
	16	1
	17	1
	18	1
	19	1
	20	1
	21	1
	22	1
	23	1
	24	1
	25	1
	26	1
	27	1
	28	1
	29	1
	30	1
	31	1
	32	1
	33	1

Table 3.29 Most parsimonious cladogram and its compatible character states found by CAFCA for Strauch's Alcidae data.

When we run an heuristic search by PAUP to find the most parsimonious cladogram for this data set, we find 11 cladograms with 64 steps (38 steps is the theoretical minimum). The number of characters states compatible with these cladograms ranges from 53 to 55 (including the 33 states compatible with the root).. None of these cladograms has taxa 18 through 21 as a sistergroup of taxa 1 trough 17, as, according to Strauch, should be the case. This grouping is present in CAFCA's most parsimonious cladogram, apparently at the cost of three extra steps.



Cladogram–1 : COMPATIBILITIES

Cladon	Character	State
23	28	2
24	33	2
25	24	2
27	2	2
28	21	2
29	20	2
30	18	2
	19	3
31	9	2
32	7	2
	8	2
	10	2
	22	3
	27	3
34	3	2
	16	2
45	5	2
	26	2
	29	2
46	22	2
	31	2

Table 3.30 One out of 11 most parsimonious cladograms found by PAUP for Strauch's Alcidae data, and its character compatibilities as listed by CAFCA.

When we analyse one of the 11 most parsimonious cladograms found by PAUP (table 3.30; 64 steps) in CAFCA as a user-tree in order to trace its compatibilities, we find only 18 *fully* compatible characters. The reason is clear. This cladogram is even more resolved, although not completely, than the ones found by CAFCA, at the cost of *fully* compatible characters, but gaining 3 extra steps. It appears that maximising the number of fully compatible characters, or compatible character states for that matter, does not necessarily lead to minimal trees, i.e., is not equivalent to minimising the number of steps (see also dePinna, 1991). Sticking to the maximal set of fully compatible characters can block the route to the most parsimonious cladograms.

The states compatible with the root node are omitted. They are the same as shown earlier in the list of compatibilities, and only relate to state 1 of all the characters as they are all directed and ordered. From the list above characters 19, 22, and 27 must be omitted as well as they are incomplete; only one of their three states is compatible with groups in the cladogram. They do not comply with the definition of a compatible character.

We can explore the relationship between character state compatibility and cladogram length in more detail by using PAUP to generate 9470 cladograms (search incomplete) with a length less than or equal to 67 steps (the restricted MPC found by CAFCA has 67 steps). The maximum number of compatible character states remains constant for cladograms in the range of 64-67 steps (figure 3.6).

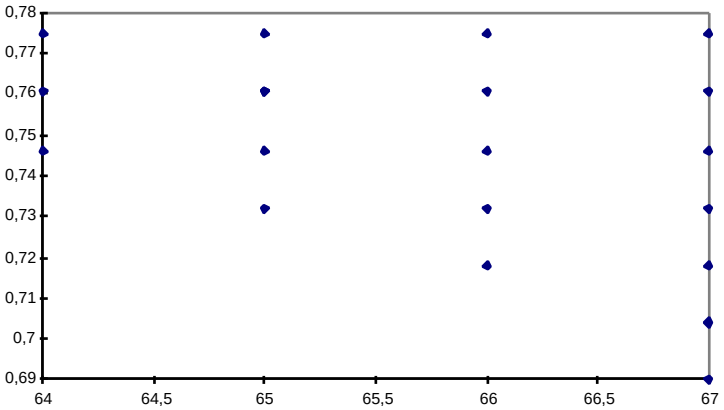


Figure 3.6 The relation between cladogram length (x-axis) and number of compatible character states as expressed by the CCSI (y-axis) in Strauch's Alcidae data.

ANDERSON'S DATA

Moving onwards on the scale of complexity we find Anderson's data (table 3.31) as an example of the use of character compatibility analysis. This matrix is more complex in comparison with Strauch's data.

It contains much more multi-state characters (with up to 6 states), and rather many homoplasies as is made clear by a parsimony analysis using PAUP that results in 1 MPT (figure 3.8) with 250 steps and a rescaled consistency index (RC) of 0.323. The theoretical minimum number of steps is 132.

```
/ G.F. Estabrook and W.R. Anderson, Syst. Bot. (1978) 3: 179 -196
/ An estimate of phylogenetic relationships within the genus Crusea
/ (Rubiaceae) using character compatibility analysis.
/                          1         2         3         4         5
/                 123456789012345678901234567890123456789012345678901234567890123456789012345678
/                 --------------------------------------------------------------------------------
coc-coccinea      111334222251321212122212314313424633224335134422121223221 1
coc-chiriquensis  111333613241441212122212314314424533224431134422121223221 1
coc-breviloba     111333321241321212122212314313324533213231123322121212121 1
megalocarpa       221134313251211212122223132131132331224332233222313342112
coronata          212311321214122122222122123233112441122324411113321222112 1
diversifolia      212111211114112122222222233132111332111114211222221222111 1
lucida            112322311231122121222222332222212321132314321341211122111 2
parviflora        114233121241112121211222431121211121121114311341212111111 2
psyllioides       212322512222122121221212332131211112111114111221322111121 1
calcicola         213412512222212122222211223231211322121114111221322112121 1
setosa            214424532233242121222211423231212212121213111221412112121 1
longiflora        222222332331212112221211323221212331222131223213121121211
calocephala       124323242333232212122121132222231323213341212243132211212 1
wrigh-wrightii    223223443332222121221112232222123221212111123132211212 1 1
wrigh-angustifolia 223213333323222121221113222222122221323111224313221121211
hisp-hispida      224234344351232121221211322232213122132311122331422112231 1
hisp-grandiflora  224234444351232121221211322322231321444111335414221122311
```

Table 3.31 Anderson's data on *Crusea*.

Running CAFCA with different options for cladon definition results in cladograms for *Crusea* that are always longer than the one found by PAUP for unordered characters. A summary of CAFCA's findings is given in table 3.32. The best cladogram found by CAFCA for PMS and ordered characters is 320

steps long (277 steps with characters unordered). It contains 3 trichotomies. When they are resolved by secondary analysis a cladogram results with 301 steps (figure 3.7; 264 steps with characters unordered).

Option	#Clada	#Cladograms	Length
PMS, ordered	111	38	320-440 277-326 (unord)
PMS, unordered	155	244	269-302
PMS + ABC	543	2541	276-317
SMS	948	>>50.000	269-...

Table 3.32 CAFCA results for cladogram lengths from the *Crusea* data when run under different options

The list of compatible character states for the CAFCA secondary cladogram (figure 3.7) is given in table 3.33. There are 12 characters, # 10, 15, 16, 19, 20, 21, 22, 37, 48, 51, 52, 57, together having 27 states, that are *fully* compatible with this cladogram (The best cladogram from the primary analysis has 11 fully compatible characters; # 51 is absent). However, in addition there are many more *individual character states* from other incompatible characters that are also compatible with this cladogram.

Estabrook and Anderson (1978) reported 4 cliques of 13 characters and 2 cliques with 12 characters each. Their cladogram based on 13 compatible characters is much less resolved than the results obtained by PAUP and CAFCA's secondary analysis. Farris and Kluge (1979) contains severe criticisms on the character compatibility method and on the many ad hoc manipulations that Estabrook and Anderson (l.c.) allow to arrive at their final solution (cladogram).

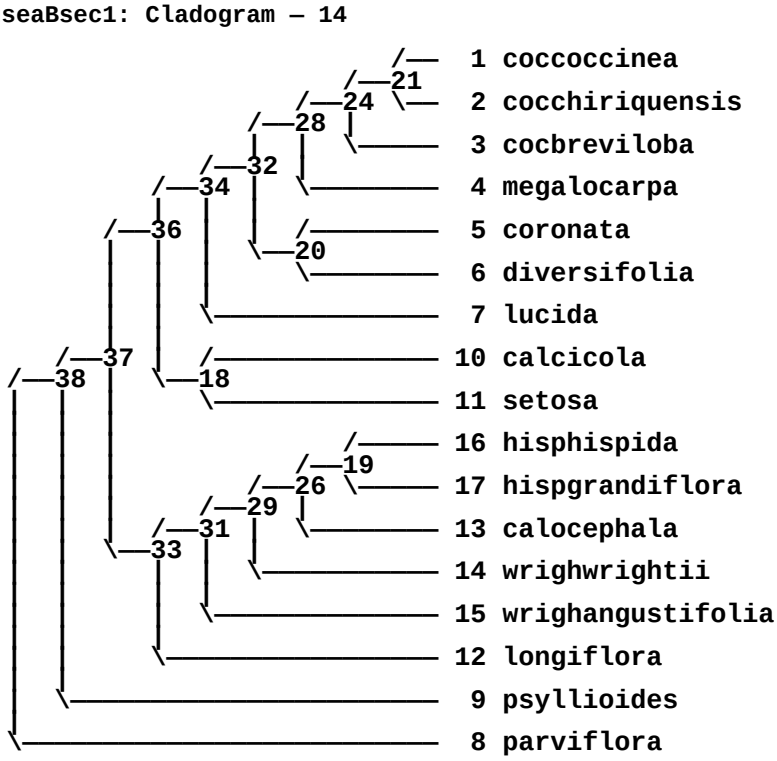


Figure 3.7 Single MPT found by CAFCA after secondary analysis, based on PMS and ordered characters.

CruseaBsec1: Cladogram–14 :
COMPATIBILITIES

Cladon	Character	State
1	34	6
	42	5
2	7	6

	13	4		14	1
	30	4		15	2
4	50	3		16	1
	52	3		17	2
	53	3		18	1
	54	4		19	2
5	35	4		20	1
	43	4		21	1
	48	3		22	2
	57	2		23	1
17	38	4		24	2
	46	5		25	3
18	4	4		26	3
19	9	4		27	1
	56	3		28	1
20	12	4		29	2
21	31	4		30	1
	45	4		31	2
24	27	4		32	1
	28	3		33	1
	33	4		34	1
	34	5		35	2
	37	2		36	1
28	16	2		37	1
	41	3		38	2
29	8	4		39	1
32	48	2		40	1
	52	2		41	1
33	10	3		42	4
36	21	2		43	2
37	54	2		44	1
38	7	3		45	1
	20	2		46	3
	27	2		47	3
39	1	1		48	1
	2	1		49	2
	3	3		50	1
	4	2		51	2
	5	2		52	1
	6	2		53	1
	7	2		54	1
	8	1		55	1
	9	1		56	1
	10	2		57	1
	11	3		58	1
	12	1			
	13	1			

Table 3.33 List of compatible character states for the single cladogram for *Crusea* found by CAFCA after secondary analysis (PMS, ordered characters).

When CAFCA is run with option 1 (PMS) and unordered characters it finds 244 cladograms five of which are MPT's with 269 steps. These MPT's all show 2 trichotomies. If we look at one of them (# 66) it has 13 *fully* compatible characters (with 32 states), i.e., 10, 15, 16, 20, 22, 24, 37, 41, 48, 51, 52, 54, and 57, and a total of 58 compatible character states. After secondary analysis this cladogram is completely dichotomous and counts 262 steps and only 12 *fully* compatible characters, but still 58 compatible character states.

PAUP can also be run under different options for character types (irreversible, ordered, unordered). The results are summarised in table 3.34

Character type	# MPT's	Length: ordered	Length: unordered	Length: irreversible
irreversible	2	292, 298	256, 259	375
ordered	1	286	255	n.a.
unordered	1	293	**250**	430

Table 3.34 PAUP results for cladogram length when run under different options for character types.

The cladogram found by PAUP on the basis of unordered characters (figure 3.8; 250 steps) has 12 *fully* compatible characters, i.e. # 10, 15, 16, 19, 20, 22, 24, 37, 41, 48, 52, 57 and a total of 41 compatible character states. This set is almost identical to the set for the CAFCA cladogram based on ordered characters except the interchange of 21 + 51 for 24 + 41. Apparently, on the basis of the same number of fully compatible characters (12) shorter cladograms (293 steps when characters are considered ordered) are possible as is shown by the PAUP analysis, but the number of character *states* that are compatible with these cladograms is very different (41 vs 58).

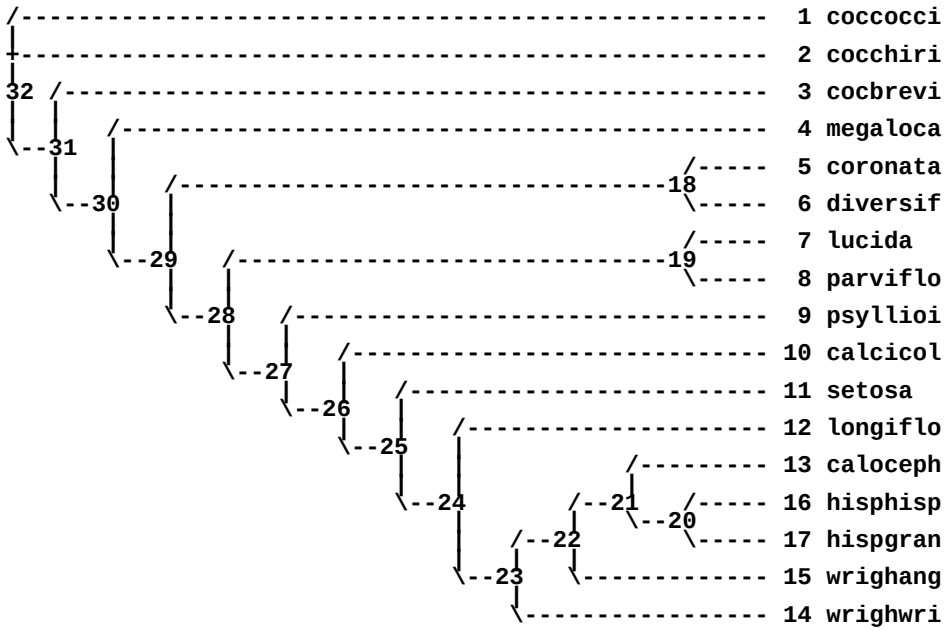


Figure 3.8 Single most parsimonious tree resulting from PAUP analysis on unordered characters in Andersons's *Crusea* data.

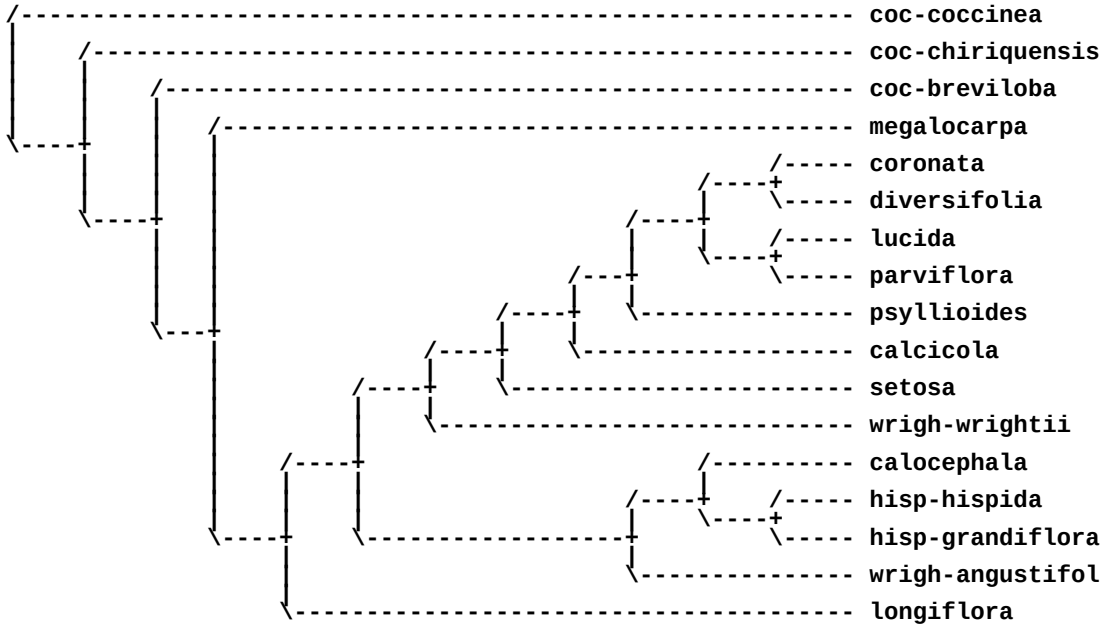


Figure 3.9 Single most parsimonious tree resulting from PAUP analysis on ordered characters in Andersons's *Crusea* data.

The PAUP cladogram based on ordered characters (figure 3.9; 286 steps) has only a set of 8 compatible characters, i.e. # 15, 16, 17, 19, 20, 22, 37, 57 (and 93 compatible character states, including 58 states on the root and 15 aut-

apomorphies). This is a subset of the set for the cladogram based on unordered characters. The CAFCA cladogram based on ordered characters is longer (301 steps after secondary analysis) but has 12 compatible characters (and 74 compatible character states, excluding those for the root). Obviously, in the CAFCA result one pays a price in steps for more character states to be congruent with the cladogram.

As in Strauch's *Alcidae* data, this example again clearly shows that holding on to the maximal set of fully compatible characters (maximal clique), i.e., constraining the analysis to find only those solutions for which the characters in the clique show no extra steps, can make it impossible to find the most parsimonious cladogram. Maximising the number of compatible characters, or compatible character states for that matter, does not necessarily lead to minimal trees, i.e., is not equivalent to minimising the number of steps.

We can use CAFCA's 244 cladograms for unordered characters and combine these with the 280 cladograms in the range of 250-256 steps as found by PAUP to visualise the relationship between number of steps and the number of compatible character states (figure 3.10). In this particular case we see two clusters of cladograms because we refrained to completely close the gap between PAUP's MPC (250 steps) and the shortest cladogram found by CAFCA (269 steps) due to the very many cladograms in the range of 256-269 steps. Also, CAFCA in this case does only find cladograms which are cliques of PMS's of terminal taxa. As a consequence many other cladograms do not obtain, especially those with a low number (< 45) of compatible character states, which leaves a blank space in the scattergram. The scattergram clearly indicates that cladograms optimal for the CCSI are not MPC's and that MPC's for complex data do not have the largest number of primary homologies. It is also clear that in general the relation between number of steps and the number of compatible character states is not linear. In the MPC many primary homologies have failed the test of congruence and must be reinterpreted as independently acquired secondary homologies.

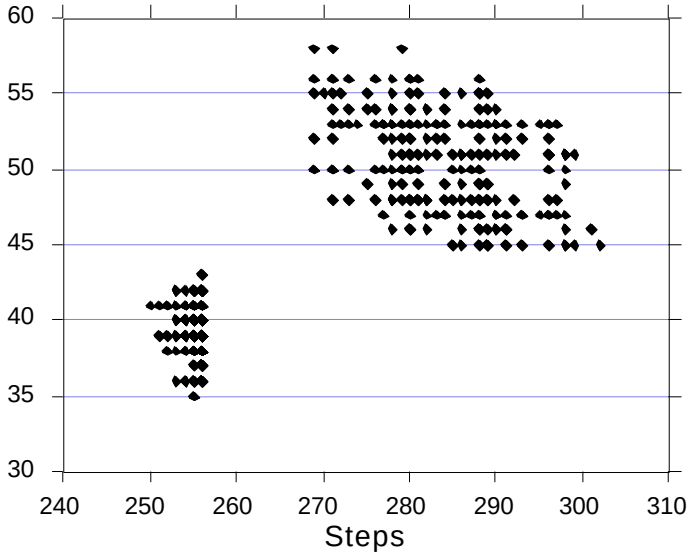


Figure 3.10 Number of steps (x-axis) versus number of compatible character states (y-axis) for 244 cladograms found by CAFCA with option PMS and unordered characters and 280 cladograms in the range of 250-256 steps found by PAUP for unordered characters. For explanation see text.

The two PAUP cladograms based on irreversible (up) characters have a set of only 5 fully compatible characters. Again, imposing order clearly increases

the number of conflicts (incompatibilities, incongruencies) among characters; the more strict the ordering the more conflicts are present.

TAXONOMIC EFFICIENCY

According to Rodrigo (1991) taxonomic efficiency (TE) refers to "the ease of which we may identify taxon membership in practice". Taxonomic efficiency is defined in terms of the number of characters that are consistent (compatible) with the cladogram and therefore mutually compatible (Rodrigo, 1991). The concept can easily be enhanced by allowing degrees of taxonomic efficiency. The most strict sense of TE is measured in terms of compatible characters. When all clades in a cladogram can be identified by at least one unique character, taxonomic efficiency is optimal (these characters are mutually compatible by definition). When not all clades have such an identifying character, taxonomic efficiency is sub-optimal. We can lift this constraint and turn to less strict taxonomic efficiency, but still adhere to the original purpose of ease of identifying taxon membership. In the less strict sense TE is measured by the number of compatible character *states* for a cladogram (as in the compatible character state index) or the set of partial or strict monothetic sets it supports. Also in the less strict sense we can distinguish optimal TE from the sub-optimal cases.

If we look at the different solutions for the *Crusea* data as well as all other examples (dePinna, Meacham, Strauch) from the viewpoint of taxonomic efficiency it is clear that irrespective of their length all cladograms found on the basis of partial or strict monothetic sets of taxa (options PMS and SMS) are efficient as **all clades** concerned can be identified either by a unique single character state or a unique combination of character states (neither one of the separate states in the combination needs to be unique).

The concept of taxonomic efficiency is not limited to the set of most parsimonious trees, as originally intended (Rodrigo, 1991). The set of taxonomic efficient cladograms obviously is much larger than the set of MPC's, and as shown in the examples so far, taxonomic efficiency in terms of either PMS's or SMS's decreases gradually as cladograms become shorter.

In some cases the MPC is sub-optimal with respect to its taxonomic efficiency in the sense that not all clades have a unique character, or not all clades are different strict monothetic sets, i.e., some are identical because characterised by the same SMS of character states. They may each have their synapomorphy, but in all these cases the synapomorphic state originates more than once independently in the MPC.

As is made clear in the examples used so far, in most practical cases the price to be paid in number of steps for a cladogram that is fully efficient taxonomically, i.e. a cladogram in which all clades have a unique character state, is often high and accumulates to a lot more steps than present in the MPC.

CONCLUSION

In the foregoing examples from the literature we have seen that CAFCA's group compatibility method in its default option (PMS) is, in general, not identical with character compatibility. For CAFCA characters need not be *fully* compatible; *CAFCA only considers compatible groups of terminal taxa based on character states and derivations thereof*. For that reason cladograms found by CAFCA may be more resolved than those built from cliques of compatible characters. In one of its other options, SMS or strict monothetic sets, CAFCA is moving even farther away from the concept of character compatibility in that

groups of taxa can be defined by ***combinations*** *of compatible character states.* To that end, character states in their original distribution over taxa may be broken down in subsets, depending on the patterns of congruence with other states in other characters. The concept of character compatibility does not account for that. Furthermore, groups of terminal taxa may be defined by character states in undirected and unordered multi-state characters, a case for which the proof of the Pairwise Compatibility Theorem, and thus its application in character compatibility analysis, is still open. Of course, this type of characters can be *shown* a posteriori to be mutually compatible, when they all fully support a cladogram, that is, have a CI=1 for that cladogram.

I explained earlier that compatibility of character states and of sets of terminal taxa is an a priori assessment based on primary homology. As a consequence the MPC may not have the largest set of compatible character states. In Meacham's data it still does, but in more complex cases like Strauch's Alcidae and Anderson's *Crusea* the cladograms with the highest CCSI can be MPC's or not.

In summary:

a. If option 1 (PMS) finds the overall MPC's than at least one of them will have the highest CCSI.
b. If primary homologies need to be broken down, like in option 2 (SMS) and 5 (TTSP), and MPC's result, than these will not have a maximal CCSI.
c. If SMS or TTSP do not result in MPC's and you find MPC's with a standard parsimony approach, than on average the MPC's will have a lower CCSI than MPC's found through b).

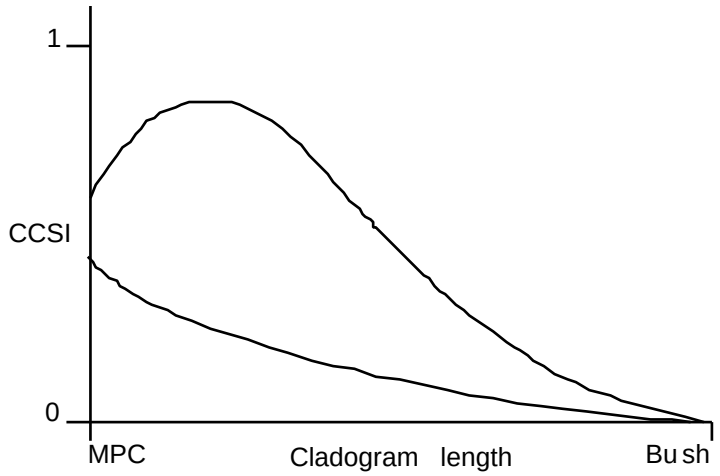


Figure 3.11 General relation between the compatible character state index and cladogram length in case of more than 1 MPC. The area enclosed by the curves is the range within which CCSI varies.

In general, when there is more than 1 MPC, the relation between cladogram length and number of compatible character states will have a form as depicted in figure 3.11.

If we want to use the results of CAFCA as a lower bound in the search for MPC's than we have the problem that we are on the top of the CCSI hill but we do not know on which side to descend (without computing cladogram length, that is) in order to find the zone of MPC's.

Although more flexible in comparison with character compatibility, it appears that *group compatibility* based on *character states or derivations thereof*

as implemented in CAFCA is not very effective nor very efficient (table 3.32) in finding most parsimonious cladograms based on complex character data when compared with strict parsimony analysis as implemented in, e.g., PAUP (table 3.34), as is clearly demonstrated in the example with the *Crusea* data.

THIS PAGE INTENTIONALLY LEFT BLANK