Computing Phylogenies
by Comparing Biosequences
Following Principles
# of Traditional Systematics

Zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
der Universität Bielefeld
vorgelegte
Dissertation
von
Georg Füllen

Bielefeld, im Juni 2000

Erstgutachter: Prof. Dr. Robert Giegerich,
Bielefeld

Zweitgutachter: Prof. Dr. Wolfgang Wägele,
Bochum

Contents

List of Figures

0.1 Summary

Phylogeny estimation, that is the inference of the evolutionary history of the various life forms (species) on earth, is a widely studied problem that is not yet solved to satisfaction. Studying the strengths and weaknesses of current methods that work on biosequence data, branch attraction phenomena due to unequal amounts of evolutionary change in different parts of the phylogeny are one major problem, placing the species that evolved fast in one part of the phylogenetic tree, and the species that evolved slowly in the other.

We improve the current state of the art by describing a way to avoid the attraction of species that evolved slowly, and hence share old ("symplesiomorphic") character states. These leftover character states have "eroded" away in the other species. They are detected using a calibrated comparison with an outgroup, and contrasted with shared novel ("synapomorphic") character states that testify the exclusive common heritage of a subset of the species. Torn apart, these shared novelties indicate conflict in a split of all species considered, and only the split at the root of the phylogenetic tree cannot have such conflict. Therefore, we can work top-down, by heuristically searching for a minimum-conflict split, and tackling the resulting two subsets in the same way. This application of the divide-and-conquer principle, together with an intelligent search for minimum-conflict splits based on the exchange of species that carry the conflict, results in a fast, simple and transparent phylogeny estimation algorithm.

The algorithm, called "minimum conflict phylogeny estimation" (MCOPE), is validated intensively using both real and artificial data. We reanalyze published trees, yielding more plausible phylogenies, and analyze small "undisputed" trees on the basis of alignments considering structural homology. Artificial data feature randomly constructed phylogenetic trees with equal and unequal amounts of evolutionary change.

Our phylogeny inference method may be viewed as a quantification of the reasoning that a systematist applies whenever s/he builds up a tree based on morphological data, using cladistic principles.

Chapter 1

Introduction

1.1 The Light of Evolution

"Nothing in biology makes sense except in the light of evolution."
Theodosius Dobzhansky, American Geneticist [5]

For a long time, humans have been eager to understand life, as well as themselves, and the roots and origin of both. Modern science places man among all life forms, and tries to describe how these first appeared and have since diversified and changed. Evolution theory describes the initial conditions and the underlying processes. We will assume that this theory has universal validity even though we note that this is impossible to prove. Moreover, many flavors of this theory exist, but we will restrict ourselves to the most basic "undisputed" principles, thereby reducing the probability that we make false assumptions.

In fact, we do not need much more than the principle of "descent with modification": Starting with a set (population) of "almost" identical ancestral life forms, we assume that a reproductive mechanism led to generations after generations of descendants, some of these modified with respect to the original. Environmental conditions and other factors separated descendants into subpopulations. After many generations, modifications accumulated such that the reproductive mechanism only works *within* the separated subpopulations, which we may then call "species". If the separation-descent-modification process occurs repeatedly, the result is a phylogeny, that is a hierarchical, tree-like structure that represents the evolution of the various species. Given the species as they are today, our task will be the estimation of the underlying phylogeny; we have to calculate the order of separation events that happened in the past.

Our emphasis will be on the evolutionary processes, that are the processes postulated by evolution theory. The initial conditions (origin-of-life issues) are not investigated in this text, nor are the structures on which they have operated and still operate. These structures, and the possibility to order their substructures sequentially, for example in the form of biomolecular sequences, are taken for granted. Furthermore, we gloss over any variation within a species. Finally, we ignore that some species carry hundreds of blueprints (genes) of some of these structures, since there are mechanisms that keep the blueprints almost identical, like unequal crossing-over and gene conversion [4]. Ignoring variation, we can take one such structure like 18S-rRNA, and talk about "its sequence" and claim that there is exactly one characteristic 18S-rDNA sequence for each species. This sequence is composed of substructures called nucleotides, like any RNA/DNA sequence is. Taking proteins, the substructures would be amino acids. (18S-rRNA is a component of ribosomes, which are part of the apparatus that translates gene information into proteins. It has its own gene, the 18S-rDNA sequence.)

Since we assume that the evolutionary processes are behind all history of life, an investigation of life forms is always an investigation into the results of evolution – "Nothing in biology makes sense except in the light of evolution" [5]. No matter what we are talking about – molecules, genomes, organisms or ecosystems – considering the history of the entities under study can help us a great deal in understanding structure, function and relationships.

In this introduction, we highlight four examples of the productive use of phylogenetic information in biology research – this is "the light of evolution" in action.

- *Molecular level.* The prediction of the three-dimensional structure of proteins can be improved significantly if we know the three-dimensional structure of related proteins. By finding these relatives, phylogeny serves as an aid for molecular modelling, see e.g. [27].

- *Genomic level.* The phylogenetic analysis of viral genes can be exemplified by the case of HIV, the human immunodeficiency virus. The possibility that a dentist has infected his patients has been studied by estimating the phylogeny of the viruses carried by the dentist and his patients [23, 11]. The global spread of HIV subtypes is studied phylogenetically in [24]. Cross-species transmission events and the discovery of recombination between viruses are examined in [30]. As a last example, phylogeny inference is used to assess the worldwide variation of HIV and some other viruses in [12]. The phylogeny of many virus families has been studied up to now; the influenza virus is another prominent example, see e.g. [18].

- *Organism level.* The development of body plans can be studied in the light of phylogenetic data. The author of [20] even expresses his hope to establish a "causal relationship" between the evolution of genes called *HOM/Hox clusters*, and the evolution of body plans. The paper already discusses a variety of correlations between gene evolution and organism development.

- *Ecosystem level.* Phylogenies are very useful for the description of biodiversity and the inference of population processes in wildlife, as described in [21]. This paper also discusses how wildlife management, and conservation in general, may benefit from studying phylogenies.

We have highlighted the path of gaining knowledge by considering the evolutionary history of the entities under study. This work is concerned with phylogeny estimation, i.e. gaining knowledge about the evolutionary history itself. The basis for gaining such knowledge is currently expanding at a rapid pace. Due to improvements in nucleotide sequencing technology, larger and larger datasets are in need of phylogenetic analysis, featuring significantly more than just 30 species and just a few hundred nucleotides/amino acids. Instead, for hundreds of species, thousands of nucleotides are now available for analysis. In fact, whole genomes are becoming available, making an all-encompassing phylogenetic analysis possible for the first time. Whole genomes comprise huge datasets on the order of billions of nucleotides, and it would be worthwhile to align the data as far as possible, and to estimate trees from the data that comprise all the inheritable information of the different species.

1.2 Minimum Conflict Phylogeny Estimation in a Nutshell

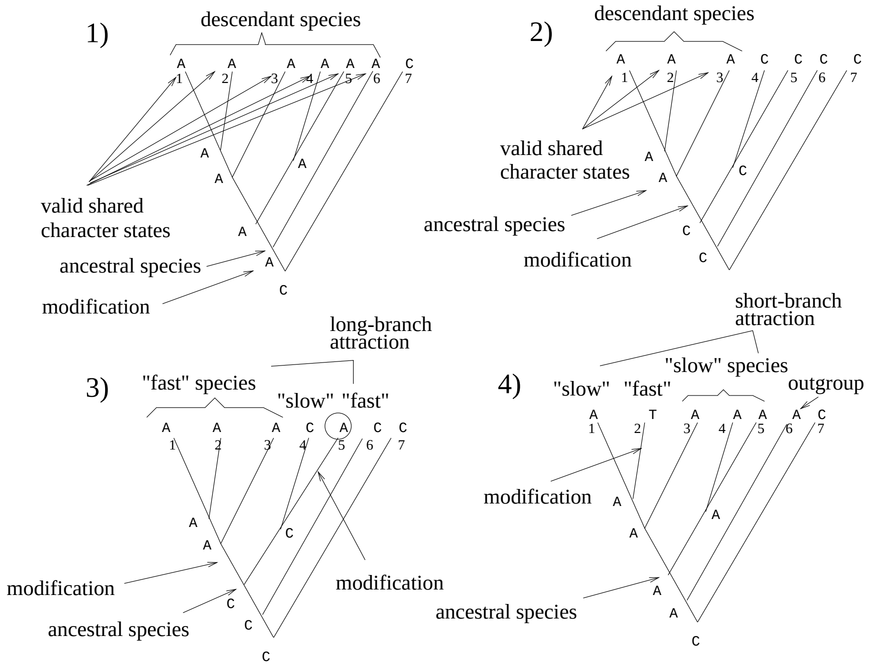
In the following, we will discuss some general characteristics of our approach. In the next section, we will contrast our approach to current methods.

The first characteristic of our approach is simplicity, and a focus on the most relevant information. We already described our simple model of evolution, which amounts to the hypothesis that separation and descent with modification are the processes that we should focus on. The most natural way of analyzing data resulting from these processes is to look for the modifications. In the case of biomolecular sequences, these are modifications of character states (sequence substructures like nucleotides) that appeared anew in an ancestral species. They testify the exclusive common heritage of all the species to which that ancestral species gave rise. If at least some of these modifications are still visible in the present-day species, we should be able to detect them.

Consequently, our method for recovering the phylogenetic tree tries to detect character states shared between species because these species are *the sole descendants* of an ancestral species. We argue that these *"valid" shared character states* constitute the most relevant and the least misleading information available. Two examples are given in Fig. 1.1, panels 1 and 2, where character states are exemplified by the nucleotide symbols `A,C,G,T`.

All too often, it happens that character states are shared between some species even though there is no common ancestor from which only these species developed. Instead, these species just feature a different amount of evolutionary change. During their evolution, they were subjected either to much more, or to far less modifications than the others. The former phenomenon leads to "long-branch attraction", and the latter to "erosion", or "short branch attraction". We will describe both phenomena, which are also displayed in Fig. 1.1, panels 3 and 4. (Different amounts of evolutionary change are attributed to a "different speed of evolution", and it has become standard to talk about "fast"

Figure 1.1: Valid shared character states, and branch attraction phenomena.

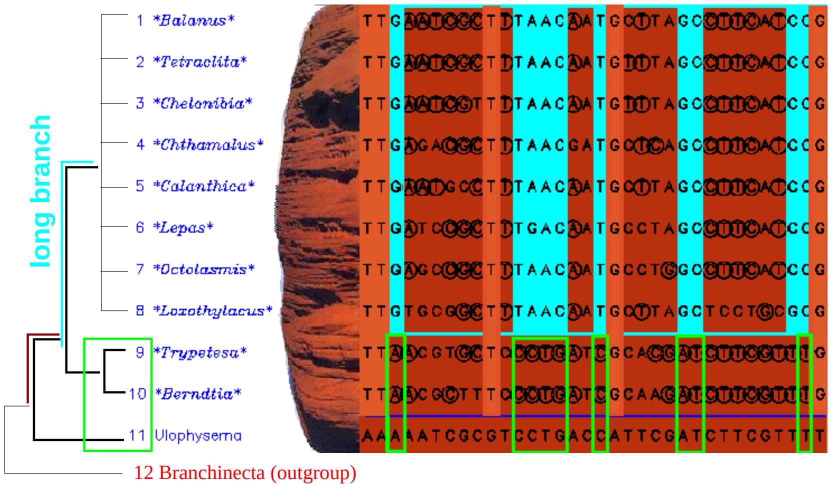


and "slowly" evolving sequences.)

"Long-branch attraction" is the observation of character states shared by "more evolved" sequences because they are modifications that are equal ("convergent") just by coincidence (Fig. 1.1, panel 3). "Short branch attraction" is due to character states shared by "less evolved" sequences because they are the leftover of old character states that were modified in the other ("more evolved") sequences only (Fig. 1.1, panel 4). Short branch attraction can be visualized by an "erosion" process taking place in the "more evolved" sequences, modifying some of their character states that have been shared before. Since there are usually (but not necessarily) at least a few modifications that coincide with character states elsewhere, at least a low level of long-branch attraction is often a phenomenon that occurs together with short branch attraction. Due to erosion, tree estimation algorithms may be misled by the similarity of the old character states still shared by the uneroded, less evolved sequences.

In general, however, long-branch attraction may happen independently of short-branch attraction, and both may happen independently of the separation-descent-modification process that underlies the phylogeny. For example, the "more evolved" sequences may share similarities due to random convergences, and the "less evolved" sequences may nevertheless have been subject to so many modifications that they feature no shared old character states. A more problematic theme, to which we will return repeatedly in this text, is the possibility that branch attraction happens in parallel with the separation events, that is in concordance with the evolutionary history of the species.

Figure 1.2: Erosion in species 1-8 triggers artifact similarity in species 9-11.


long branch
1 *Balanus*
2 *Tetraclita*
3 *Chelonibia*
4 *Chthamalus*
5 *Calanthica*
6 *Lepas*
7 *Octolasmis*
8 *Loxothylacus*
9 *Trypetesa*
10 *Berndtia*
11 Ulophysema
12 Branchinecta (outgroup)
TTGAATCGCTTTAACAATGCTTAGCCTTCATCCG
TTGAATCGCTTTAACAATGTTTAGCCTTCATCCG
TTGAATCGTTTTAACAATGTTTAGCCTTCATCCG
TTGAGACGCTTTAACGATGCTCAGCCTTCATCCG
TTGAATGCCTTTAACAATGCTTAGCCTTCATCCG
TTGATCCGCTTTGACAATGCCTAGCCTTCATCCG
TTGAGCCGCTTTAACAATGCCTGGCCTTCATCCG
TTGTGCGGCTTTAACAATGCTTAGCTCCTGCGCG
TTAACGTGCTCCCTGATCGCACGATCTTCGTTTG
TTAACGCTTTCCCTGATCGCAAGATCTTCGTTTG
AAAAATCGCGTCCTGACCATTCGATCTTCGTTTT

A likely case of short-branch attraction is exemplified in Fig. 1.2, which displays an alignment of 18S-rDNA from crustacean species on the right, and the putative correct tree on the left. The ancestor of species 1-8 gained many modifications compared to the other species. This erosion in species 1-8 triggers shared old character states in the short-branch species 9-11, in contrast to the valid shared character states that testify the exclusive common heritage of species 1-10. In contrast, the shared old character states do not testify exclusive common heritage of species 9-11 – these species do not have a common ancestor that is not the ancestor of any other species. (It does *not* matter along which branch(es) the larger amount of evolutionary change took place, as long as all species in 1-8 are affected. For example, if they are only recent subjects to significant modifications, we can still observe erosion. In other words, the "caves" in evolved species may consist of conserved modifications, gaps and/or a mix of nucleotides.)

Fig. 1.2 introduces our running example, to which we will return throughout this text. It is based on real data, and the putative correct tree is derived from morphological features. A partial alignment will provide us with data for sample calculations. The full-length alignment will be discussed extensively in section 4.4.1.

Short-branch attraction can be detected by inspecting a sequence or sequences from species that developed as a side-branch from an ancestor "older" than the ancestral species giving rise to all the descendant species under consideration. This "outgroup" (species 6 in Fig. 1.1, panel 4, and species 12 in Fig. 1.2) indicates whether character states shared by the short branches only might be old indeed. If they are, we can disregard all artificial evidence that places the short-branch species into one group, and the long-branch species into the other. The decision "old" versus "new" is based on the calculation of "matching rates" with respect to the outgroup. A matching rate compares two sequences (or sets of sequences) and it is calculated by tallying the number of character states that are equal. (For sets of sequences, majority character states are checked for equality.) Basically, the more matching with the outgroup we observe, the more evidence we have for erosion. Matching rates are a very simple concept that we will use again and again; after all, simplicity and transparency shall be one important characteristic of our approach. When the analysis is started, the user will need to specify an outgroup in advance. However, we will show in section 3.3 that we have an unusual freedom in the choice of the outgroup, because we will calibrate the matching rate of the shared character states by comparing it with a matching rate tallied over alignment columns that do not feature the shared character states.

To summarize, the second, and possibly the most important characteristic of our method is that it tries to detect short branch attraction – it avoids falling into what we call the erosion trap, and to our knowledge it is the first method that explicitly avoids this systematic error. We even conjecture that as long as the short-branch attraction is stronger than the long-branch attraction artifact, we can detect the former and then avoid *both* problems. This is "Future Work", see section 4.6.

The description above is idealistic because usually, there is no clear-cut division between "more evolved" and "less evolved" sequences. Moreover, the speed of evolution may differ in time, across the branches of the tree, resulting in a complex mixture of branch attraction phenomena, and possibly other artifacts. Nevertheless, our method is able to recover correct separations in many cases, as described in the sections on validation.

Once we are able to ignore shared old character states, and instead concentrate on the valid ones that indicate *the sole descendants* of an ancestral species, we can do a *heuristic search* of all splits (bipartitions) of the set of species analyzed. Starting with any split, "conflict" may arise if valid shared character states can be found for a subset of the species: if this subset is torn apart by the split, the valid shared character states are then found on both sides. They are torn apart themselves. If a split with no (or minimum) conflict can be found by moving species between the two sides, we assume that we have found the most ancient separation. In hindsight, this most ancient separation divides the set of species into two subsets, and every subset includes *the sole descendants* of one of the two ancestral species into which the species at the starting point (at the so-called root) was separated.

Now, our approach can make use of the divide-and-conquer paradigm enabling the fast analysis of large datasets. We have already motivated the necessity of processing speed in section 1.1, and we will now show that divide-and-conquer is a natural ingredient of our approach.

The heuristic search just discussed is designed to reveal the most ancient separation, and the question is how the analysis can be continued. The most natural answer is to use *divide-and-conquer.* We view the two separated sets of species as new problems that can be tackled in the same way. Indeed, in a top-down manner, we will explore the hierarchical structure of the dataset by estimating the most ancient separation, followed by the analysis of the two subsets that result from the corresponding split. The two subsets are then analyzed in exactly the same way as the whole dataset was analyzed before; only the outgroup may be different. It is selected in a way that ensures the most informative matching rates. Once the subsets are analyzed, we follow up on their minimum-conflict splits, and so on, until the analysis stops with sets of one or two species. The divide-and-conquer scheme is visualized in Fig. 2.7 on page 36.

For a completely balanced tree of 100 species, the first divide-and-conquer step divides the problem into two sets of 50 species each. Next, we need to tackle four sets of 25 species, etc. For unbalanced trees, divide-and-conquer slows down, reducing a problem with 100 species to a new one with 99 species, etc. We assume that trees arising from samples of existing species are usually rather balanced. Evidence for this assumption is as follows:

- There is no need to question the overall validity of classic biological systematics, where species are actually classified into many *large* groups of related species on different levels of a classification hierarchy.

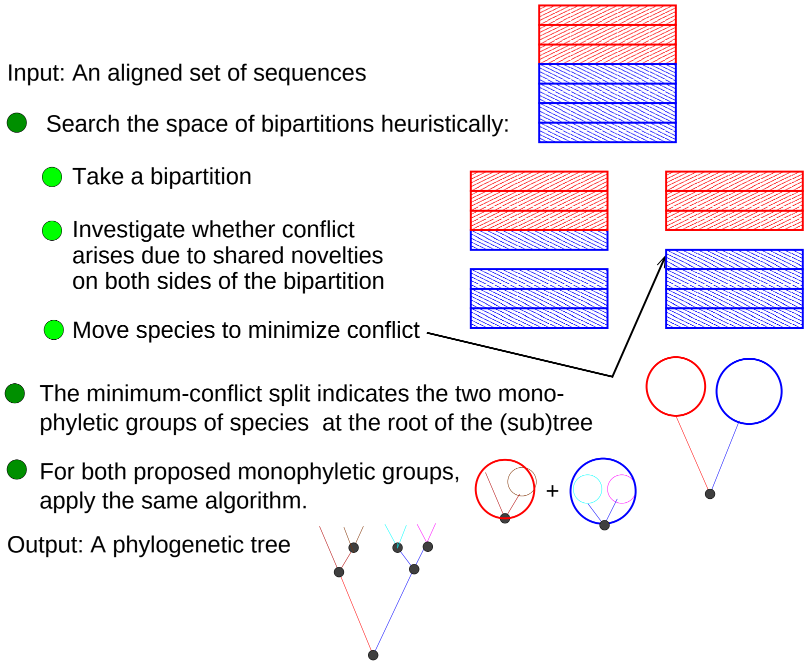
- Trees published in the literature are usually quite balanced; few are completely imbalanced so-called "caterpillars" (cf. Fig. 3.29, panel 1, on page 103).

- Simulation studies usually suggest even more balanced trees, and this "puzzle" is the subject of some recently revived research (see e.g. [1]).

All trees considered are rooted, making both erosion detection and divide-and-conquer possible. These are the third and forth characteristic of our method: We can expect that the algorithm is fast, and we calculate rooted trees (in contrast to less informative unrooted trees) as a side-effect.

A high-level overview of our method, termed "minimum conflict phylogeny estimation" (MCOPE) can be found in Fig. 1.3. In general, we aim to model the decision process of a trained systematist who applies a strictly logical approach to phylogeny estimation. Our method may be viewed as followup work that builds upon the logics of phylogeny inference, based on the concepts of *shared novelties* (synapomorphies), *convergences* (homoplasies), and *leftovers* (symplesiomorphies), see e.g. [10] and sections 2.6 and 2.8. The relevance of this “cladistic” approach developed by Willi Hennig has been outlined before [41, 42]. We attempt to quantify it, and improve on certain aspects. For example, sigmoid functions are used repeatedly to achieve discriminatory power, e.g. to amplify and filter the evidence found via the comparison of matching rates.

Figure 1.3: Schematic overview of MCOPE.



Input: An aligned set of sequences

- Search the space of bipartitions heuristically:
  - Take a bipartition
  - Investigate whether conflict arises due to shared novelties on both sides of the bipartition
  - Move species to minimize conflict
- The minimum-conflict split indicates the two monophyletic groups of species at the root of the (sub)tree
- For both proposed monophyletic groups, apply the same algorithm.

Output: A phylogenetic tree

The fifth characteristic of our approach is the transparency that comes with its logical foundation: We record the evidence for different hypotheses of phylogenetic relationship, analyse and compare it using simple formulas, and make it possible for the researcher to re-evaluate both the evidence and its analysis.

We believe that it is very important to validate a new method for gaining phylogenetic knowledge. Validation with biological data (in contrast to artificially generated data) is important to prevent circularity, which may occur in subtle ways whenever likeminded researchers write data generation as well as data analysis software. We are tempted to say, "Nothing in phylogeny estimation is validated except in the light of biological data." On the other hand, in the case of artificial data we can be sure to know the correct phylogeny. Therefore, we have done an extensive validation by biological data and artificial data alike. Applied to both kinds of data our method performs very well. In particular, we will discuss several examples where evidence from molecular datasets is now much more in line with morphology-based systematic knowledge.

1.3 Comparison with Other Phylogeny Estimation Methods

In this section, we compare our method to other approaches. For a detailed explanation of these, and a comprehensive overview of phylogenetic systematics in general, the reader is asked to consult [36].

Distance methods for phylogeny estimation calculate distances between *pairs* of sequences, based on the number of character states that do not match. The pairwise distances are used to build up a phylogenetic tree. Our method, however, is character-based, analyzing multiple sequences simultaneously in a position-by-position fashion. This makes more use of the information provided by the individual sequence positions. Furthermore, modifications may occur in the development of the present-day species only, after the last separation event. These amplify distances between close relatives, and misguide the analysis. In an attempt to deal with this specific type of long-branch artifact, corrections of distance estimates are often employed, and they are obtained by using specific models of character state evolution.

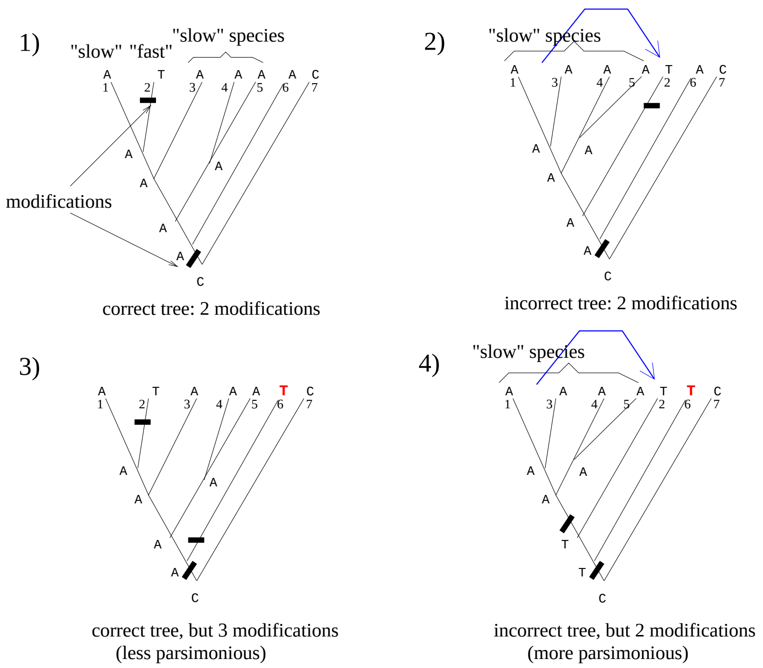
Such detailed models of evolution, including specific "substitution rates" for different classes of modifications, cannot be universally valid, and their estimation from data analyzed before constitutes circular reasoning: what if the trees used to estimate the model parameters are incorrect ? In this case, a systematic error is introduced into the model, and it may be reinforced by the further analyses of similar data.

Like our approach, *maximum likelihood* analyses are character-based. These evaluate a so-called likelihood function for each sequence position, and combine the results. Likelihood gives high scores to trees for which the modifications conform to an estimate about which classes of modifications are likely, and which ones are not. Therefore, the likelihood function relies on a detailed model of character state evolution, which we would like to avoid for the same reasons as in the case of distance methods. Furthermore, noise caused by random modifications may nevertheless influence the result of maximum likelihood.

Parsimony analysis is also character-based, evaluating another (but similar, see [38]) scoring function for each sequence position and combining the results. Parsimony gives high scores to trees which explain the data with a minimum of modifications. No detailed model is needed for the parsimony function, but a lot of the noise caused by random modifications may influence the result of parsimony. If positions are weighted differently (see [36], pp. 502-503, for a review), and/or the correction suggested by [29] is applied, the problem may at best disappear at the expense of additional complexity that allows for other systematic errors. Most importantly, we now discuss how parsimony may be mislead by erosion.

Parsimony can fall into the erosion trap because trees for which the short branches form a subtree require less modifications, if the long branches match the character state estimated for the node at which they are all attached. This matching may be viewed as a low level of long-branch attraction, and we con-

Figure 1.4: Parsimony may be misled by erosion.



correct tree: 2 modifications

incorrect tree: 2 modifications

correct tree, but 3 modifications
(less parsimonious)

incorrect tree, but 2 modifications
(more parsimonious)

jecture that pure short-branch attraction cannot mislead parsimony.

In Fig. 1.4, panels 1 and 2 feature the same character states at the leaves of the tree. The difference is in the tree itself: species 2, representing ”the long branches” in this case, is attached to different edges, and we assume that the left-side tree is correct. In panels 3 and 4, the character state of species 6 is T, but the tree in panel 3 is the same as in panel 1, and tree in panel 4 coincides with the one in panel 2. Inferred modifications are marked by thick lines. If species 2 has a character state that is different from species 6, we need two modifications to explain its evolution, no matter which tree we take (panel 1 or panel 2). If species 2 has a character state matching species 6, the correct tree on the left requires three inferred modifications (panel 3), and parsimony will favor the incorrect tree where species 2 and species 6 are ”together”, and there is no need for a modification ”between them” (panel 4). In the most parsimonious tree, the short-branch species 1,3-5 form one subtree. For panel 3, we remark that three modifications are also needed if we assume that the last common ancestor of 1-6 has character state T. Moreover, there is more than one tree that is both incorrect as well as most parsimonious; species 2 and 6 in the tree to the right (panels 2 and 4) may as well be placed in a common subtree, and the number of modifications is two for such a tree as well.

In contrast to all three standard methods, our method builds the tree top-down, from the root to the leaves. At each level we do a simultaneous analysis of the relevant sequences only. As we have seen, this strong focus has a very pleasant side-effect: at least for balanced trees, calculations can be very fast.

Of the existing approaches just discussed, parsimony and maximum likelihood in particular are reaching their limits for large datasets, especially because these require a (heuristic) search of the space of all possible trees, evaluating the scoring function very many times. Researchers are reworking these methods in order to make them faster. For example, "PUZZLE" [33] is a method that does maximum likelihood calculations for sets of 4 species, and then assembles the subproblem solutions recursively. For parsimony, "Iterative Fixation" [29] is one algorithm that combines species into archetypes to speed up calculations. However, these two approaches are limited by both the drawbacks of the underlying method, and the heuristic nature of the speedup.

Then again, the divide-and-conquer paradigm already plays a key role in tackling large phylogenies. A distance-based generic algorithm called "Disk-Covering" is described in [13, 14]. "Disk-Covering" is similar to our approach because it also divides the set of species into subsets, and then combines the subtree solutions. Its reliance on distance calculations makes it susceptible to the problems already discussed for distance trees. Nevertheless, "Disk-Covering" holds a lot of promise, in particular because it has proven desirable characteristics for sufficiently large sequences, see the end of section 2.3.

1.4 Nature, Models, and Software

In the following chapter, we will introduce our basic concepts and then we will formalize the "phylogeny estimation" problem that we are trying to solve. Later we will define the "minimum conflict" method that we propose to estimate phylogenies, and our final task will be the attempt to validate our method. This section discusses some issues in formalizing biological problems, and using computers to solve them.

Strictly speaking, we need to distinguish 3 different sets of entities: nature, models, and software. The first set of entities is "taken from nature"; we believe that we are observers of individuals, species, reproduction, nucleotide sequences, vertebrates, evolution, and such.

Since these natural entities tend to be elusive, the next sections will develop a second set of entities. These are (simplified) models of some of the natural entities. We will define this small world of entities as precisely as possible and we will express both our problem and our method in terms of these. The problem will be stated by describing an extremly simple model of the process of sequence evolution. This process transforms a certain kind of tree into an aligned set of sequences. We will state that this aligned set of sequences evolved according to the tree, and our task will be to recover the tree, given the aligned set of sequences. The task is tackled by our method of phylogeny estimation. It will be described as a series of calculations which attempts to transform the final data of an individual process of sequence evolution, that is an aligned set of sequences, into the tree with which the process was started. We assume that there is some homomorphy (i.e. structural similarity) between the natural and the model entities and their relationships, even though we make no attempt to identify it precisely; formalizing this homomorphy is a very complex subject indeed.

Our method has been implemented on a computer, yielding yet another set of entities expressed "in silico". We have strived to faithfully map our model entities to computer code such that our method can be executed on electronic data files (input, an aligned set of sequences), and we get the same results (output, a tree) as if we executed our method on paper.

Assuming finally that the data files are faithful observations of the natural entities investigated (like nucleotide sequences aligned with the help of structural information, see below), our computations can be interpreted as the inference of natural phenomena – then, if all our assumptions are met, we are able to make statements e.g. about the evolutionary history of vertebrates by estimating a phylogenetic tree.

If such statements are plausible (i.e. not in conflict with "most" of the other observations that we make), or, possibly, even yield correct predictions in the world of natural entities, then we "are on the right track" towards validation. While such validation is difficult, another form of validation can be obtained readily even though its value is limited. We can simply test whether our method correctly recovers the underlying tree, if it is given the results of an artificial process of sequence evolution. Indeed, we have used both "biological" and "artificial" datasets in our validation attempt; in both cases we obtain very good results, as described later.

Let us expand on the term "nucleotide sequences aligned with the help of structural information". Our method does not work with unaligned sequences, such that we assume the existence of a black-box alignment preprocessor in case of unaligned input. For ribosomal RNA sequences, the preprocessor may take into account RNA structure information; for coding DNA, protein structures may be used. In general, alignment is a tough problem, and if we have sequences which are aligned by some unknown or imperfect procedure, failure of validation using biological datasets may be attributed to an incorrect alignment. The reader may consult [7] and references therein for further information on the multiple alignment problem.

1.5 Conventions Used in This Text

Great care has been taken to use `precise language`, by defining all relevant concepts at least verbally. The first occurance of an important concept is written in `typewriter` font, accompanied by its definition. If a verbal definition is given, the reader may watch for a definition in mathematical terms soon thereafter, again quoting the concept in typewriter font. In the rare case that a concept is mentioned before it is defined, it is enclosed by quotation marks. All defined concepts are listed in the index, which gives the reader a look-up glossary.

Chapter 2

Analyzing Patterns of Evolution

2.1 Species, Character States, and Alignments

We will now define a small set of entities that will be used to give a precise problem statement, and, later on, a precise description of the algorithm we propose. Along the way, we develop not only models for the biological entities we work with, including a very general model of evolution. We also formalize some cladistic concepts that are used very frequently by biologists to describe and analyze the results of evolutionary processes. Regarding both the general model and the formal treatment of cladistic terminology, we are not aware of similar efforts.

The first entity to be defined is a "species". In the spirit of the introductory statements, we evade the issue of what a species really is. For us it is just a name, and we understand that it refers to a population of life forms that do not reproduce with other life forms. All data considered is from recent species, that is from entities which are alive today, as opposed to dinosaur or other fossile sequence data.

We are given m names that we call `recent species`, or `species` for short, per default denoted by indices $1, ..., m$. We assume that we have m sequences, $s'_1, ..., s'_m$, one sequence per species, each `sequence` consisting of `character states` taken from an `alphabet of symbols` $\mathcal{A}$, e.g. $\mathcal{A} = \{\texttt{A},\texttt{C},\texttt{G},\texttt{T}\}$ for DNA sequences.

In the course of its evolution, a sequence may be extended or shortened. We assume that we also have an `alignment` S of the sequences (S is also known as the `data`), a matrix consisting of m rows and r columns with character states taken from the extended alphabet of symbols $\mathcal{A} \cup \{-\}$, where "$-$" $\notin \mathcal{A}$ denotes the `special character state` "gap":

$$\begin{array}{lllllll} s_1 = & s_{1,1} & s_{1,2} & \dots & s_{1,j} & \dots & s_{1,r} \\ s_2 = & s_{2,1} & s_{2,2} & \dots & s_{2,j} & \dots & s_{2,r} \\ & & \dots & & & & \\ s_i = & s_{i,1} & s_{i,2} & \dots & s_{i,j} & \dots & s_{i,r} \\ & & \dots & & & & \\ s_m = & s_{m,1} & s_{m,2} & \dots & s_{m,j} & \dots & s_{m,r}. \end{array}$$

The sequences ${s'}_1, ..., {s'}_m$ can be obtained from the rows (the aligned sequences, denoted $s_1, ..., s_i, ..., s_m$) of S by removing all gaps. The alignment S must not have any columns (sites, denoted $s_{*,1}, ..., s_{*,j}, ..., s_{*,r}$) that consist exclusively of gaps.

Two character states are matching if they are equal, or equivalent (i.e. in the same equivalence class, given an equivalence relation on the set of character states). Character states which match are also called shared, those that do not match are called different. In a fixed (or invariable) alignment column, all species have matching character states. The notion of "matching" may be generalized to "similarity". This is useful in the case of amino acid character states, and for morphologically defined ones, e.g. features of specific body parts of animals. We will only deal with RNA/DNA sequences, and there will be no follow-up on this generalization.

The alignment of all sequences to be analyzed is always denoted by S, and it has r columns; alignments in general are denoted by A. Unless noted otherwise, calculations done with an alignment A ignore its fixed columns, pretending that the alignment just consists of variable sites. The projection $A_{|_I}$ of any alignment A to a set of rows indexed by a subset I of the species,

$$I = \{i_1, ..., i_\ell\} \subseteq \{1, ..., m\}$$

is called a subalignment; its columns are called subcolumns, and its length (ignoring gap-only columns as well as fixed columns) is denoted by q. Subsets of subsets of species will be denoted by

$$I' = \{i_1, ..., i_k\} \subseteq \{i_1, ..., i_\ell\}.$$

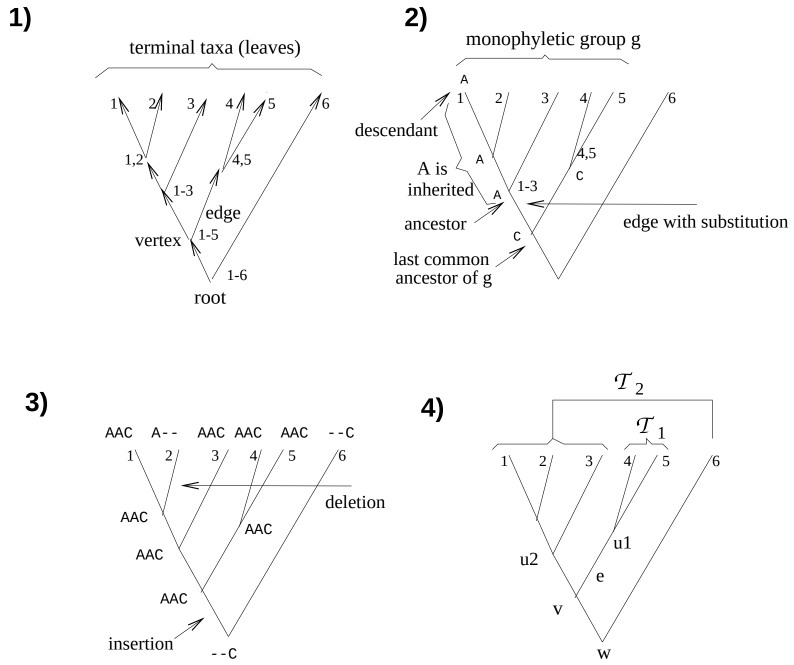
The projection $A_{|_{I'}}$ has length p. (The indices are selected such that their lexicographic order reflects their size, $k \leq \ell \leq m$ holds as well as $p \leq q \leq r$.)

2.2 Trees and Monophyly, Heirs and Ancestors

The entities defined in this section prepare the way for a formal model of evolution. Trees are used as the formal representation of the hierarchy that is created if descent with modification is interspersed with separation events.

A group of species g is just a set of species. A subgroup is a subset, and a supergroup is a set that includes the group under consideration. Given a group

Figure 2.1: Tree terminology and examples.



of species g and a subgroup g' of g, the `complementary subgroup` is denoted by $g - g'$.

A `split` $G = g1$ v $g2$ (read "$g1$ versus $g2$") is a bipartition of a group of species into two nonempty subgroups, $g1$ and $g2$.

A `tree` of m species is a directed cycle-free connected graph $\mathcal{T} = (V, E)$ with vertices V and edges $E = V$ x V, which has m `terminal` vertices with one incoming edge, one `root` vertex with two outgoing edges, and a set of non-root internal vertices with one incoming and two outgoing edges (see Fig. 2.1, panel 1, where $m = 6$). A tree is also known as a `phylogeny`, or a `phylogenetic tree`. The vertices are also known as the `nodes`, the terminal vertices are the `leaves`, and the edges are also known as the `branches` of the tree. All edges are directed from the root to the terminal vertices, and every non-root vertex is a `direct descendant` : it has one incoming edge from its `direct ancestor`. We label the terminal vertices with the species indices $\{1, ..., m\}$, the root vertex with the description of the list of all species, that is the single label "1-m", and any other internal vertex v with the description of the list of terminal labels that can be reached from v by travelling away from the root. In formal terms, these `heirs` are defined by

$$heirs(v) = \{i \in \{1, ..., m\} : \exists u_1, u_2, ..., u_z \,:\, \{(v, u_1), (u_1, u_2), ..., (u_z, i)\} \subset E\}\,.$$

Vertices u_* in the above definition describe the path from v to i. The symbol $\exists$ should be read as "there exists". The set of all heirs of v is the set of vertices i for which there exists a path from v to i. As a generalization of the preceeding definitions, a path from v to i makes v an ancestor of i, and i a descendant of v. The heirs of v are all descendants of v. A vertex v is a common ancestor of a set of species $\{i_1, ..., i_k\} \subseteq \{1, ..., m\}$, if

$$\{i_1, ..., i_k\} \subseteq heirs(v)$$

holds.

All trees considered in this text are rooted and bifurcating. We note that in practice, we may consider multifurcating trees (with vertices of degree 4 or more, so-called polytomies). On the one hand, there are some cases where multifurcations may have happened in the natural world of entities. On the other hand, multifurcations can express uncertainties in estimated trees, allowing to represent several hypotheses in a single tree.

For an internal vertex v the label lists the monophyletic group of all the species for which v is the last common ancestor. More generally, for any group of species $\{i_1, ..., i_k\} \subseteq \{1, ..., m\}$, the last common ancestor $lca(i_1, ..., i_k)$ is defined as

$$\begin{aligned} lca(i_1, ..., i_k) = v \iff \quad & v \text{ is a common ancestor of } \{i_1, ..., i_k\}, \text{ and} \\ & \forall v' \neq v : \left(\{i_1, ..., i_k\} \not\subseteq heirs(v') \text{ or } heirs(v) \underset{\neq}{\subset} heirs(v') \right) \end{aligned}$$

The second condition states that all vertices that are not the last common ancestor are either missing one or more descendants, or they are the ancestor of a strictly larger group of species. (The symbol $\forall$ should be read as "for all".)

The last common ancestor v leads to the two last common ancestors of the two disjoint sister groups that the monophylum splits into (see Fig. 2.1, panel 2. The last common ancestor of g = "1-5" leads to the last common ancestors labeled "1-3" and "4,5". "1-3" is the sister group of "4,5" and vice versa. We will discuss some aspects of panel 2 as well as panels 3 and 4 of this figure in the next sections.) The closer to the root, the older an ancestor is.

2.3 Evolution and Phylogeny Estimation

Given a formal definition of trees, we can now write down a formal model of evolution. Let us first add another label to each vertex of our tree $\mathcal{T}$. For the terminal vertices, we use the aligned sequences of the given alignment S of the m species. For the internal vertices, we use some arbitrary aligned sequences of same length. Then, such a tree represents a specific evolutionary history of the m sequences, and its interpretation is as follows.

- Every internal vertex v corresponds to a separation event, that is the verbatim copying of its aligned sequence.

- Every edge corresponds to zero or more `substitutions` in the aligned sequence, that is the modification of a character state into a different one.

We assume that the character states of an alignment column all evolved from a single character state of the root ancestor. Then, a `character` is just an alignment column. If a character state does not change across an edge (or a path of edges), we say that the descendant `inherited` it from its ancestor (see Fig. 2.1, panel 2, on page 20).

The reader may ask which "natural entities" are modeled by the aligned sequences. In biological terms, every aligned sequence represents a species comprising individual life forms. These natural entities are subject to selection, genetic drift, reproduction, and speciation, but no precise definition of these biological terms will be given in this text. The model that we will describe will subsume these effects, however.

Let $z = |\mathcal{A} \cup \{-\}|$ denote the size of our alphabet, including the gap character. Substitution frequencies across an edge e can be recorded by a z times z `substitution frequency matrix` M, where the entry $m_{x,y}$ is the relative frequency of a substitution of the xth symbol by the yth symbol, for $x \neq y$, and the entry $m_{x,x}$ is the frequency that the xth symbol is not substituted. Naturally, we have

$$\sum_{x \in \{1,\ldots,z\}} m_{x,y} = 1 \text{ for all } y,$$

and

$$\sum_{y \in \{1,\ldots,z\}} m_{x,y} = 1 \text{ for all } x.$$

In the case of nucleotide sequences, the matrix M is

$$\begin{aligned} M &= \begin{pmatrix} m_{1,1} & m_{1,2} & m_{1,3} & m_{1,4} & m_{1,5} \\ m_{2,1} & m_{2,2} & m_{2,3} & m_{2,4} & m_{2,5} \\ m_{3,1} & m_{3,2} & m_{3,3} & m_{3,4} & m_{3,5} \\ m_{4,1} & m_{4,2} & m_{4,3} & m_{4,4} & m_{4,5} \\ m_{5,1} & m_{5,2} & m_{5,3} & m_{5,4} & m_{5,5} \end{pmatrix} \\ &= \begin{pmatrix} m_{\mathtt{A}\to\mathtt{A}} & m_{\mathtt{A}\to\mathtt{C}} & m_{\mathtt{A}\to\mathtt{G}} & m_{\mathtt{A}\to\mathtt{T}} & m_{\mathtt{A}\to-} \\ m_{\mathtt{C}\to\mathtt{A}} & m_{\mathtt{C}\to\mathtt{C}} & m_{\mathtt{C}\to\mathtt{G}} & m_{\mathtt{C}\to\mathtt{T}} & m_{\mathtt{C}\to-} \\ m_{\mathtt{G}\to\mathtt{A}} & m_{\mathtt{G}\to\mathtt{C}} & m_{\mathtt{G}\to\mathtt{G}} & m_{\mathtt{G}\to\mathtt{T}} & m_{\mathtt{G}\to-} \\ m_{\mathtt{T}\to\mathtt{A}} & m_{\mathtt{T}\to\mathtt{C}} & m_{\mathtt{T}\to\mathtt{G}} & m_{\mathtt{T}\to\mathtt{T}} & m_{\mathtt{T}\to-} \\ m_{-\to\mathtt{A}} & m_{-\to\mathtt{C}} & m_{-\to\mathtt{G}} & m_{-\to\mathtt{T}} & m_{-\to-} \end{pmatrix}, \end{aligned}$$

treating the gap symbol as the fifth nucleotide character state. If $m_{x,x} > \frac{1}{z}$ holds for character state x, we can still see "traces" of inheritance that are preserved for that state. If inheritance is traceable for all character states, we say that the substitution frequency matrix still reveals a minimum of inheritance. The closer $m_{x,x}$ is to $\frac{1}{z}$, and the shorter the aligned sequences, the more likely any small inheritance observed must be interpreted as an artifact, the result of chance alone.

A `deletion` is represented by the substitution of one or more consecutive nucleotides into gap character states. An `insertion` is represented by the substitution of one or more consecutive gap character states into nucleotides. In other words, our representation of evolution accomodates insertions in an indirect way, using gap character states in aligned ancestral sequences as a placeholder for future events (see Fig. 2.1, panel 3 on page 20). An `indel` is either an insertion or a deletion.

Up to now, we have been talking about the *representation* of a *specific* evolutionary history, given m species and an alignment, by a tree labelled with all the aligned sequences, both ancestral and recent. We did not use the term "model" because we would like to reserve it to the "evolutionary process" in general, of which a specific evolutionary history is just a sample result.

First however, we need to specify the following `evolutionary parameters` if we want to give a complete specification for a model of sequence evolution.

- A family of `character probability distributions` that is used to create the aligned root sequence. Formally, we define a family of functions
$$f_j : \mathcal{A} \cup \{-\} \rightarrow [0..1]$$
where
$$\sum_{\mathtt{N} \in \mathcal{A} \cup \{-\}} f_j(\mathtt{N}) = 1,$$
for each site j of the root sequence. For each site j, a different distribution may be used for the creation of the root character state.

- A family of `substitution probability distributions` that is used to apply substitutions across the edges of the tree. Formally, we define a family of functions
$$g_j : E \rightarrow \underbrace{[0..1]^z \text{ x ... x } [0..1]^z}_{z\ times}.$$
Function g_j assigns a `substitution probability matrix`
$$M^j(e) = \{m^{j,e}\}_{x \in \{1,\dots,z\}, y \in \{1,\dots,z\}}$$
to each edge $e \in E$, where
$$\sum_{x \in \{1,\dots,z\}} m^{j,e}_{x,y} = 1 \text{ for all } y \in \{1,...,z\},$$
$$\sum_{y \in \{1,\dots,z\}} m^{j,e}_{x,y} = 1 \text{ for all } x \in \{1,...,z\},$$
and
$$1 > m^{j,e}_{x,x} > \frac{1}{z} \text{ for all } x \in \{1,...,z\}.$$
For each site j, and for each edge e, a different matrix may be used. The third condition ensures in particular that a minimum of traceable inheritance takes place along an edge.

Our peculiar way of using substitution probability matrices is possible because we have one matrix per edge; as soon as we insist on a universal substitution probability matrix, we need to introduce a notion of time into our model if we want to have "slow" and "fast" evolution along different edges, and we then need to deal with multiple substitutions across a single edge, e.g. from `A` to `G` and back to `A`. Then, we are forced to define "instantaneous rate matrices" Q that refer to the amount of change for an "infinitesimally" small amount of time, and solve the differential equation to arrive at a matrix that describes the rate of change for an arbitrary period of time t. (The reader may be familiar with the corresponding Jukes-Cantor matrices Q_{JC} and M_{JC} in the case of 4 nucleotides (see e.g. [36]):

$$Q_{JC} = \left(\begin{array}{llll} -3\alpha & \alpha & \alpha & \alpha \\ \alpha & -3\alpha & \alpha & \alpha \\ \alpha & \alpha & -3\alpha & \alpha \\ \alpha & \alpha & \alpha & -3\alpha, \end{array} \right),$$

and, solving a differential equation,

$$M_{JC} = \left(\begin{array}{llll} \frac{1}{4}+\frac{3}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} \\ \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}+\frac{3}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} \\ \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}+\frac{3}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} \\ \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}-\frac{1}{4}e^{-4\alpha t} & \frac{1}{4}+\frac{3}{4}e^{-4\alpha t} \end{array} \right),$$

where α is a parameter that combines base frequency and substitution rate: it is one quarter of the mean substitution rate.)

For each edge e of the tree we can define its `inherent edge weight`,

$$\lambda_e = \frac{\sum_j \frac{\sum_{x\in\{1,\ldots,z\}} 1-m^{j,e}_{x,x}}{z}}{r},$$

that is the expected substitution probability on that edge, given a random sequence at the incoming node. It is the average over all sites of the average probability that a character state is substituted at that site, which in turn is the average over all character states x of $1 - m^{j,e}_{x,x}$. (As before, z is the size of the alphabet, matrix M has z rows and z columns, and r is the length of the alignment.)

To obtain an appropriate notion of indels, restrictions would need to be imposed on the probability distributions such that creation and substitution of gap characters as well as substitution *by* gap characters tend to be done along consecutive stretches of the sequence – we will not describe these restrictions in detail, however.

Given m species, an alignment length r, a tree $\mathcal{T}$ and evolutionary parameters $\{f_j\}$ and $\{g_j\}$, an `evolutionary process` is the generation of the aligned sequences S with the evolutionary parameters $(\{f_j\},\{g_j\})$ along the tree $\mathcal{T}$. Strictly speaking, this is our very simple `model of sequence evolution`.

Our model is sufficiently general that we may have specific character-dependent substitution probabilities, different substitution rates on different branches of the tree, sequence motifs (regions in the alignment where the substitution probability is low), and many more features. In fact, our model is too permissive for any practical considerations, unless we impose larger lower bounds on $m^{j,e}_{x,x}$. The current bound

$$1 > m^{j,e}_{x,x} > \frac{1}{z} \text{ for all } x \in \{1, ..., z\}$$

just guarantees that $0 < \lambda_e < 1 - \frac{1}{z}$ holds for the inherent edge weights, and for inherent edge weights close to $1 - \frac{1}{z}$, the generated data are "almost" random. Larger lower bounds on $m^{j,e}_{x,x}$, which translate into tighter upper bounds on the inherent edge weights, are necessary if we want to have any chance to recover the tree, but we will not develop any theoretical results in this text – this is "Future Work". As outlined before, our model shall suffice for the aim of describing and justifying our method of phylogeny reconstruction. Then, obtaining plausible trees is the ultimate test for the usefulness of our method and the adequacy of our model.

In summary, the design of our model follows a very simple notion of evolution, "descent with modification": Character states are subject to substitutions, and inherited by descendants. We ignore reticulate (net-like) evolution with confounding factors like lateral transfer of genetic information between some species, e.g. in bacteria, or plants.

Now we are able to define the problem we want to solve, the `phylogeny estimation problem`.

Given m species and their alignment S that is the result of an evolutionary process with unknown (but fixed) evolutionary parameters $(\{f_j\}, \{g_j\})$ along an unknown (but fixed) tree $\mathcal{T}$ and $\epsilon > 0$, recover the tree with probability at least $1 - \epsilon$.

We may make the problem even more difficult, by setting bounds on the time complexity of the calculation (e.g. that we must recover the tree in time that is polynomial in both m and r).

The constant ϵ above may be arbitrarily small, but non-zero. Both the required sequence length as well as the time complexity of the method may be made dependent on ϵ. For example, both may be polynomial in $1/\epsilon$.

We do not suggest that our method will solve the phylogeny estimation problem. Instead, our aim is a method that can recover the underlying tree for as many biologically realistic evolutionary processes as possible, i.e. for as "large" and "realistic" a parameter space as possible. In particular, the lower bound on $m^{j,e}_{x,x}$, which translates into an upper bound on the inherent edge weight λ_e, must not be too small, because then the evolutionary process is just producing random data. On the other hand, if the $m^{j,e}_{x,x}$ are too large (i.e. if they are very close to 1), there are very few substitutions – not much detectable evolution is taking place. Both scenarios are not in the application domain of our method.

The reader may think that our problem cannot be solved anyway, unless the evolutionary parameters are "very favorable". However, if we forget about locating the root of the tree, and about indels, and if we restrict the phylogenetic tree to a so-called "CF tree", with two character states and an underlying Markov process with a single substitution probability matrix, where edge weights follow a Poisson distribution, and if the character states of the root sequence follow the uniform distribution, there exist methods which recover the tree in polynomial time, from an alignment of polynomial size. One of these methods is the Disk-Covering Method that we mentioned in section 1.3. The details of these results can be found in section 7.1 of [44], including the formal framework needed to state them in precise terms, and to prove them.

In practice, an algorithm for solving the phylogeny estimation problem may indicate those cases where the data are not informative, e.g. by returning multifurcating trees. In the case of our approach, we can use multifurcations to note our failure to obtain exactly one minimum-conflict split with no close followups, cf. section 2.10.

2.4 Majority Sequences

Our method will construct "majority sequences" which are related to the notion of consensus sequences. As we will see in section 2.8, majority sequences are important to identify incorrect splits. They form yet another set of labels of the internal vertices of the phylogenetic tree. The `majority symbols` are the boldfaced equivalents of the character state symbols, taken from the alphabet $\boldsymbol{\mathcal{A}} = \{\mathbf{A},\mathbf{C},\mathbf{G},\mathbf{T}\}$. A special symbol **!** denotes a position where no majority can be found: the position is called `noisy`, a term which also covers positions dominated by gaps. Formally, `majority sequences` are just sequences over the alphabet $\{\mathbf{A},\mathbf{C},\mathbf{G},\mathbf{T},\mathbf{!}\}$.

Let an alignment $A = (a_i)_{i\in\{i_1,...,i_\ell\}} = a_{i_1},...,a_i,...,a_{i_\ell}$ be given, where $\{i_1,...,i_\ell\} \subseteq \{1,...,m\}$. Then, for a symbol $\mathtt{N} \in \mathcal{A}$, let

$$\sigma(\mathtt{N}, j) = |\{i \in \{i_1,...,i_\ell\} : a_{i,j} = \mathtt{N}\}|$$

be the `character state count` of N in column j.

Given a `minimum invariability threshold` τ, $0 \leq \tau \leq 1$, $(a_i)_{i\in\{i_1,...,i_\ell\}}$ gives rise to a majority sequence

$$c((a_i)_{i\in\{i_1,...,i_\ell\}}),$$

by absolute or relative majority voting. In the case of $\tau \geq 0.5$, we use absolute majority voting:

$$c((a_i)_{i\in\{i_1,...,i_\ell\}}) = (c_{j_1},...,c_{j_q}), \text{where}$$

$$c_j = \begin{cases} \mathbf{N} & \text{if there exists } \mathtt{N} \in \mathcal{A} \text{ such that } \sigma(\mathtt{N},j) > \tau\cdot\ell, \\ \mathbf{!} & \text{otherwise} \end{cases}$$

for each column j of the alignment, where q is the length of the alignment A currently under consideration. If $\tau \geq 0.5$ does not hold, the symbol obtaining the highest vote can be selected (relative majority voting). In this case, ties are possible; these are resolved by resorting to the lexicographic ordering of the symbols in the alphabet. To formalize this notion, let $ord : \mathcal{A} \rightarrow \{1, ..., |\mathcal{A}|\}$ be an injective function which implements a lexicographic ordering of the alphabet, like $ord(\mathtt{A}) < ord(\mathtt{C}) < ord(\mathtt{G}) < ord(\mathtt{T})$. Then, the formula for c_j is

$$c_j = \begin{cases} \mathbf{N} & \left[\begin{array}{l} \text{if there exists } \mathtt{N} \in \mathcal{A} \text{ such that } \sigma(\mathtt{N}, j) > \tau \cdot \ell, \\ \text{and } \sigma(\mathtt{N}, j) \geq \sigma(\mathtt{N}', j) \text{ for all } \mathtt{N}' \neq \mathtt{N} \\ \text{and } ord(\mathtt{N}) < ord(\mathtt{N}') \text{ for all } \mathtt{N}' \neq \mathtt{N} \text{ for which } \sigma(\mathtt{N}, j) = \sigma(\mathtt{N}', j) \end{array} \right. \\ ! & \text{otherwise} \end{cases}$$

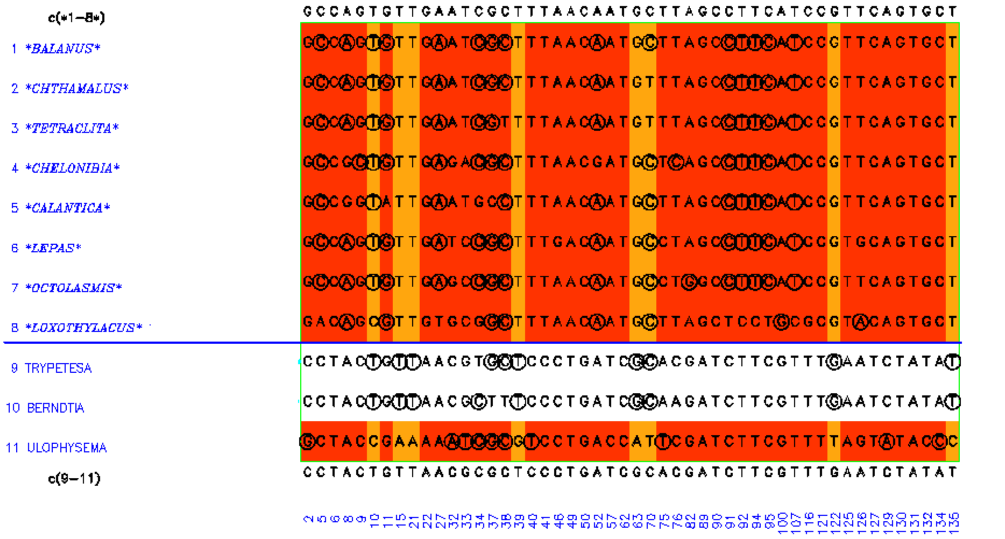
The first condition for returning **N** in the preceeding formula is a sufficient majority for **N**. The second condition ensures that no other character state has a larger vote, and the third condition takes care of the ties. If there are no gap character states in A, a minimum invariability threshold τ of 0 yields the same majority sequence as any threshold within the open interval $]0, 1/|\mathcal{A}|[$. For a bipartition (split) of an alignment A with majority sequence $c(A)$, the majority sequences of the two subalignments $A_{|g1}$ and $A_{|g2}$, $c(A_{|g1})$ and $c(A_{|g2})$, can be different from $c(A)$, and from each other.

Also, majority sequences calculated for monophyletic groups are not necessarily equal to their ancestral sequences. The ancestral character states may simply be lost in the recent species due to substitutions. Moreover, majorities are directly dependent on the choice of voters; they can be manipulated by many closely related species that are not representative of the monophylum which they "dominate". Since our method will make intensive use of majority sequences, this highlights the importance of species sampling (see also [22]); the suitable choice of species makes phylogeny estimation much easier. We will discuss in section 3.9 why our approach is nevertheless very robust as far as imbalanced species sampling is concerned. Empirically, the relative majority rule established in case of $\tau = 0$ triggers the best results; for minimum τ, a maximum of information can then be extracted and used to find "inconsistency patterns" as described in the following sections.

Running Example, Crustacean alignment

Consider Fig. 2.2 for an example. Here, an alignment of 11 sequences is displayed in a shaded box, and the split (bipartition) between sequences 1-8 and 9-11 is considered. The species names ("1 Balanus", ..., "11 Ulophysema") are listed on the left, and the column numbers (2, ..., 135) are listed at the bottom. The alignment is just the first 50 *variable* columns of the Crustacea alignment discussed in section 4.4.1; columns containing gaps or unknown nucleotides / missing data have also been removed. We already used this alignment to illustrate Fig. 1.2. (Unknown nucleotides are those which could not be identified precisely by the investigator. In the examples presented, we will always ignore columns that contain unknown nucleotides, as well as those for which data are missing.)

Figure 2.2: Majority sequences for species 1-8 and 9-11.



For the split 1-8 v 9-11, the species belonging to the first group are marked by "*". The majority sequence of the first group ("c(*1-8*)") is printed just above, and the majority sequence of the other group ("c(9-11)") is printed just below these sequences, for $\tau = 0$. (We will return to this figure in the following sections, and pay attention to more details.)

2.5 Correct Split, Missplit, Unbacked Split and the Correct Tree

In this section, we will introduce some more tree terminology that will help us describe our approach. Mostly, we are concerned with "truth": we discuss correct splits and correct trees. First, let us recall the definition of "split" given in section 2.2 : A `split` $G = g1$ v $g2$ (read "$g1$ versus $g2$") is a bipartition of the set of species into two nonempty groups, $g1$ and $g2$.

A `tree split` is the partition of a tree $\mathcal{T}$ into two nonempty subtrees,

$\mathcal{T} = (\mathcal{T}_1, \mathcal{T}_2)$, obtained by removing a single nonterminal vertex v, and one of its outgoing edges, $e = (v, u_1)$. Vertex u_1 becomes the root of the new tree $\mathcal{T}_1$. To establish the other tree $\mathcal{T}_2$ containing w, the former ancestor of v, we remove the other edges of v and add one edge connecting w with u_2, the other vertex for which v was an ancestor (see Fig. 2.1, panel 4, on page 20).

For each non-root split of a tree, there is exactly one corresponding split of the set of its species, but not vice versa. Removing the root vertex and any of its two outgoing edges gives rise to the *same* split of the set of species. This split $G^{++} = g^{++} \text{ v } \overline{g}^{++}$ at the root vertex is the only split that involves two monophyla: it involves the largest monophyletic group g^{++} and its monophyletic complement, and it is called the `correct split` of the tree $\mathcal{T}$. The `missplits` $G = g \text{ v } \overline{g}$ are the other (non-root) splits of $\mathcal{T}$; only one group in a missplit is still monophyletic, and the other group is "paraphyletic". In Fig. 2.3 on page 30, the correct split is displayed in panel 1, and a missplit is displayed in panel 2. The `unbacked splits` are all the other splits of the set of species into two groups, none of which are monophyletic, and none of which correspond to a subtree, see panel 2 (above, right). Correct split and missplits form the set of `tree-backed splits`, while missplits and unbacked splits are called the `incorrect splits` of the set of species, cf. Fig. 2.3, panel 5. Without loss of generality, we will usually assume that the first group g in a missplit $G = g \text{ v } \overline{g}$ is the monophylum.

We will also talk about the `correct split` of a monophyletic set of species, and this will be the correct split of the subtree of these species, if they are displayed in the context of a larger tree. Then, the correct split of the set of species is not the correct split of the tree displayed.

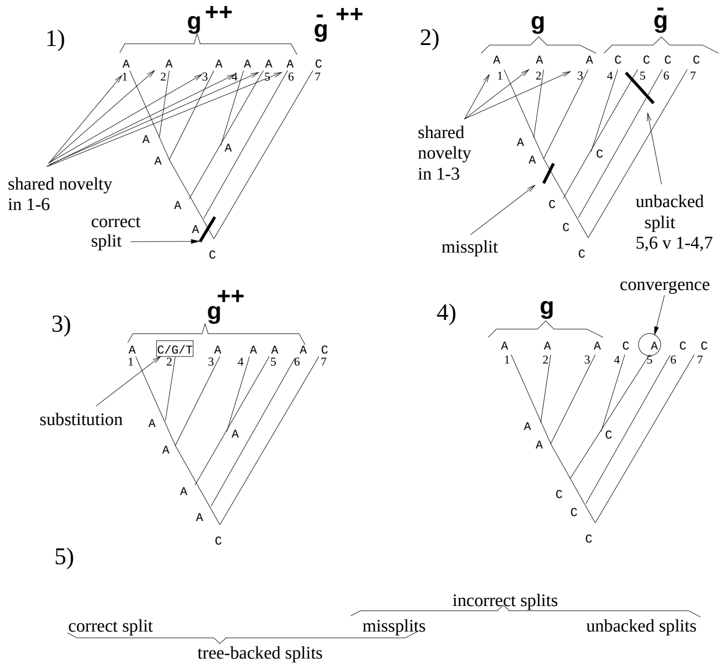
Given a set of species, we can single out one tree that we term the `correct tree`, or `most-plausible tree`, or `true tree` of the species. Its interpretation shall reflect the evolution of the species by separation and substitution alone, as defined in section 2.2. (Remember that species are just names used as labels for the sequences, and that we identify one species with exactly one sequence.) For biological data, the correct tree is extracted from observations made on natural entities, and as such, it is a problematic concept. Usually it is based on the body of knowledge assembled by systematists, employing morphological, paleontological and molecular observations. Eventually, its choice is arbitrary – ultimately it cannot be verified since we cannot look back in time. Furthermore, such a tree may not exist if our simple model of evolution is inadequate. While some inadequacies are easy to deal with (e.g. we can use multifurcating trees instead of bifurcating trees without much problem), other inadequacies are more challenging. For example, modelling and dealing with reticulate (net-like) evolution is beyond the scope of this text, and it is listed in section 4.6 on "Future Work". For artificial data, the correct tree is known by the method that was employed for generating the data.

All tree-related definitions we introduced suggest that we are dealing with the correct tree. However, terms like "monophyletic" and "correct split" are always defined with reference to a particular tree $\mathcal{T}$. Whether or not that tree is the correct tree of the species, these terms just refer to assumptions that are implicit in the tree $\mathcal{T}$, no matter what their biological plausibility is.

In the following sections, whenever we are given a tree $\mathcal{T}$, we are automatically given all its labels, including all the ancestral sequences. These make it possible to give precise definitions of the concepts involved. Nevertheless, the input of our method will only be the aligned sequences of the recent species.

2.6 Shared Novelties, Convergences, Erosion and Accumulation

Figure 2.3: Novelty, substitution and convergence.



We now define concepts and terms useful to describe the analysis of the results of evolutionary processes. Some of these are borrowed from "cladistics". For us, cladistics covers the work started and inspired by Willi Hennig (see e.g. [10]). Since the usual cladistic terms are not enjoyed by many people in the bioinformatics community, we use some more intuitive terms.

Given a tree $\mathcal{T}$, and given a monophyletic group g of species, a `shared novelty` `n` in g is a character state `n` that first appeared as a substitution in the last common ancestor of g, and was then inherited by at least two species in g.

A shared novelty is marked in Fig 2.3, panel 2 (right). Fig. 2.3, panel 1 (left) displays another shared novelty; it supports the correct split of $\mathcal{T}$. A shared novelty is also called a `synapomorphy`, or a `shared derived character state`. It is the formalization of the "valid shared character state" mentioned in the introductory sections.

A shared novelty `n` is `completely visible`, or `visible` for short, if it is present in all species of g and in no species outside g. A shared novelty `n` is `weakly visible`, if it is present in all species of g and in no species of the sister group of g. Both shared novelties in Fig. 2.3 are visible.

It follows from our model of evolution that visible shared novelties are the currency of phylogeny estimation – one single visible shared novelty about which we have absolute certainty testifies one monophyletic group; it is enough to infer one vertex (separation) of the corresponding tree. As we will see in the context of our divide-and-conquer algorithm, weak visibility is in fact sufficient for correct tree estimation.

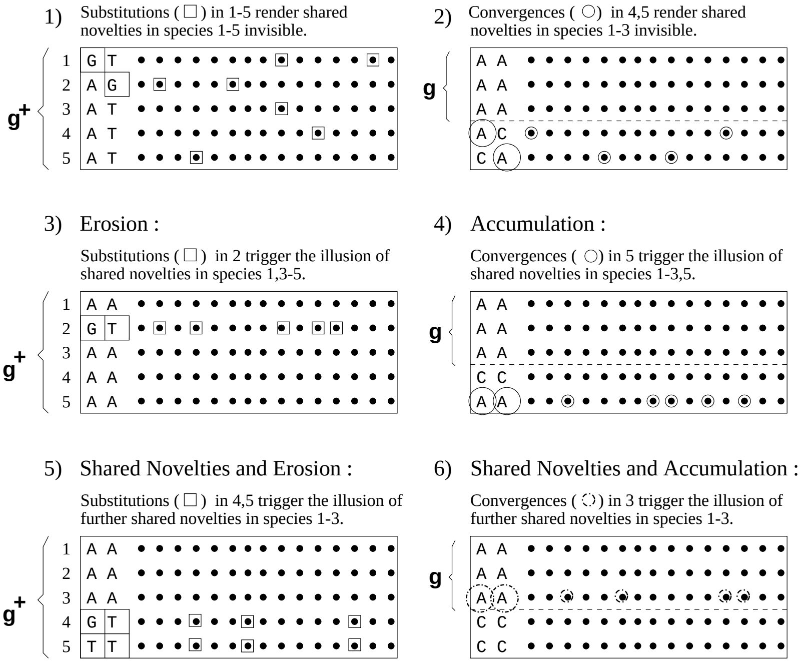
Unfortunately, ancestral sequences and inheritance relationships between character states are usually unknown, and, for biological data, there are often only few (if any) shared novelties that are visible. In particular, a big challenge for phylogeny estimation using molecular data is that shared novelties do not usually appear as insertions into the sequence, i.e. they do not usually stand out, aligning with gap characters. They are instead substitutions in alignment columns that already display a shared novelty that appeared earlier. Not only is the earlier shared novelty then made invisible, but the phenomena discussed next may lead us on the wrong track: we start seeing shared novelties that aren't there.

Remember from section 2.2 that a substitution is a character state change across an edge of the tree under consideration. One such substitution is marked in Fig. 2.3, panel 3 on page 30. To discuss invisible and imagined shared novelties in more detail, we need to distinguish convergent and nonconvergent substitutions.

Given a tree $\mathcal{T}$ and a shared novelty `n` found in a group g of species, a `convergence` to `n` is a character state that first appeared in the last common ancestor of a group of species g' disjunct from g, was then inherited by at least one species in g', and is matching with the character state of the shared novelty, cf. Fig 2.3, panel 4 on page 30. (In the figure, g' consists of the single species 5.) A convergence is also called a `homoplasy`, a `chance similarity`, or an `analogy`. It is obvious that a convergence is a substitution, but not every substitution is convergent to some shared novelty `n`. Nevertheless, convergences can be very frequent especially in molecular sequences.

Consider the group of species $g^+ =$ 1-5 in Fig. 2.4. This time, the sequences are displayed from left to right, and the species labels (from 1 to 5) are on the left. In panel 1, the sequence of the first species starts "`GT`", the sequence of the second species starts "`AG`", etc; the remaining symbols of each sequence are represented by •. We assume that the correct split into two subgroups is $g^+ = g \text{ v } \overline{g} =$ 1-3 v 4,5. We now investigate how incorrect subgroups in 1-5 may be supported by shared character states that are displayed in columns subjected to substitutions. In our search for true shared novelties, convergences – and substitutions in general – can each cause two kinds of errors.

Figure 2.4: Invisible and imagined novelties.



1) Substitutions (□) in 1-5 render shared novelties in species 1-5 invisible.

2) Convergences (○) in 4,5 render shared novelties in species 1-3 invisible.

3) Erosion : Substitutions (□) in 2 trigger the illusion of shared novelties in species 1,3-5.

4) Accumulation : Convergences (○) in 5 trigger the illusion of shared novelties in species 1-3,5.

5) Shared Novelties and Erosion : Substitutions (□) in 4,5 trigger the illusion of further shared novelties in species 1-3.

6) Shared Novelties and Accumulation : Convergences (◌) in 3 trigger the illusion of further shared novelties in species 1-3.

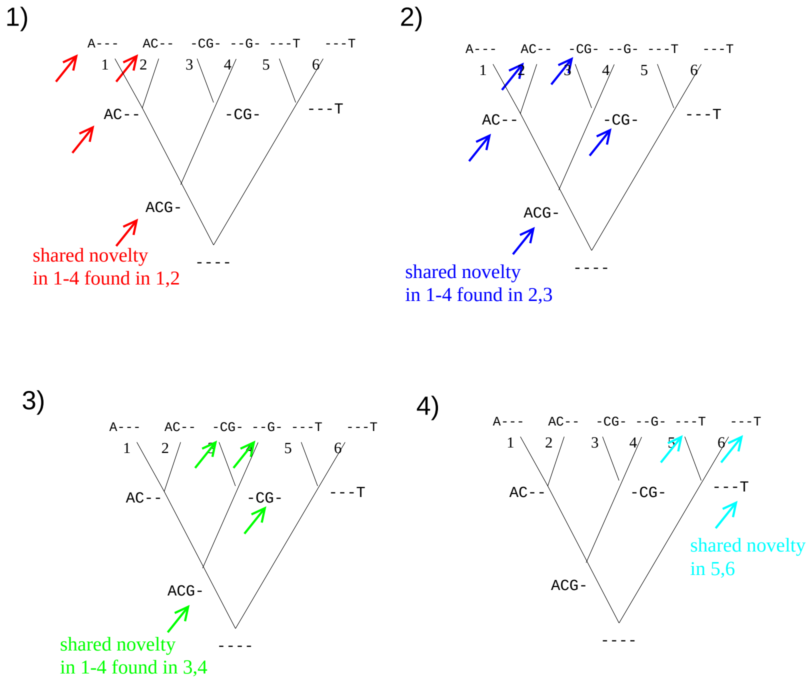
- They render existing shared novelties `invisible`:
  - Shared novelties in g^+ are subject to substitutions, see Fig. 2.4, panel 1.
  - Convergences occur in species that do not belong to the monophylum g. Convergences are the result of those substitutions outside $g \subset g^+$ that happen to coincide with the character state of the shared novelty in g, which is then invisible (panel 2).
- Their concerted action triggers the `imagination` (or `illusion`) of a shared novelty although there is none:
  - Old character states are `eroded` by substitutions in rapidly evolving species; then the other species in g^+ have matching old states, which look like a unique shared novelty (panel 3).
  - Convergences `accumulate` in rapidly evolving species; these species and the monophyletic group $g \subset g^+$ then have matching, seemingly unique novel character states (panel 4). These character states are not shared novelties because they are of different origin: their pattern does not form a "monolithic" whole – it has no unique source.

Formally, an `invisible shared novelty` is a shared novelty that does not comply with the definition of a "visible shared novelty". An `imagined shared novelty` in $g^* \subset g^+$ consists of character states that are matching because some of them are convergent to the others, or because they are leftovers of a shared novelty that was novel for the last common ancestor of a supergroup $g^+ \supset g^*$.

In fact, following up on the formal definition of a shared novelty, we have just revisited short-branch attraction (erosion) and long-branch attraction (accumulation) as discussed in the introduction (see 1). Imagined shared novelties are "false positives" in search for shared novelties, due to branch attraction phenomena.

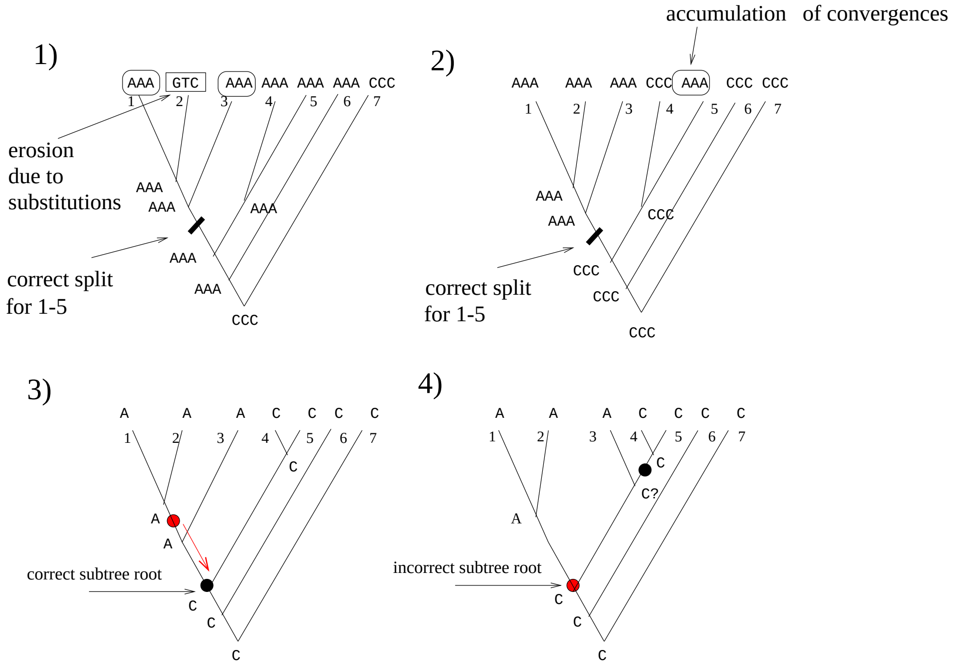
Unfortunately, the identification of shared novelties is complicated further by the possiblity that they can be found in parallel, that is, in the same species as their imagined counterpart. In other words, it may just be that branch attraction phenomena are in concordance with the evolutionary history of the species, as shown in panels 5 and 6 of Fig. 2.4. On the left, erosion in species 4,5 triggers imagined shared novelties in 1-3, indistinguishable from true shared novelties in the same species that testify the separation of their ancestor from the ancestor of 4,5. On the right (panel 6), accumulation of convergences in species 3 to shared novelties of species 1,2 triggers further imagined shared novelties in species 1-3.

Figure 2.5: Pairwise shared novelties



Our definition of shared novelty is driven by the criterion of usefulness; character states that are only inherited by at most one species do not qualify because they do not even contribute "partial knowledge" to phylogeny estimation, since they do not support groups of two or more species. Since at least 2 species display a shared novelty, we can at least place these 2 species together. Imagine the artificial scenario that some bacteria researcher was able to inhibit the preservation of any substitution in more than 2 species of his/her evolving bacteria, for some set of 6 species. Given enough shared novelties, we would still be able to establish the correct split, by combining our knowledge. In the example of Fig. 2.5, shared novelties in species 1-4 that are found in species 1,2 (red arrows, panel 1), species 2,3 (blue arrows, panel 2), species 3,4 (green arrows, panel 3) and a shared novelty in species 5,6 (cyan arrows, panel 4) would testify that the correct split is 1-4 v 5,6. Furthermore, our definition of shared novelty hints at the key idea that we develop to estimate phylogenies: shared novelties can "be torn apart", i.e. an incorrect split may feature the novel character state in both of its groups, and then we may be able to note a "conflict".

Figure 2.6: Erosion and accumulation (top); Correct and incorrect rooting (bottom).



Erosion and convergence accumulation can only be identified if they happen for many characters simultaneously. We will resort to the notion of a "novelty estimate" which serves as a quantitative measurement that is the higher the less evidence for erosion is observed.

In a tree setting, both erosion and convergence accumulation are displayed once more in Fig 2.6, panels 1 and 2. Here, 3 columns of the corresponding alignment are shown on the same tree; "`AAA`" denotes the 3 character states of the first row (species 1), "`GTC`" denotes the 3 character states of the second row (species 2), and so on. We will soon explain in great detail how imagined shared novelties due to erosion can be detected by outgroup comparison; this detection gives us good estimated trees in spite of long branches such as in species 2 in panel 1. Even detection of convergence accumulation (as in species 5 in panel 2) may be possible by modifying our approach, as mentioned in section 4.6 on "Future Work".

2.7 Divide-and-Conquer Phylogeny Estimation

Without the complications just discussed, one weakly visible shared novelty can be observed for every separation event, and the following `ideal divide-and-conquer phylogeny estimation` can calculate the correct tree, given a set of species G:

- Find the shared novelty involving the largest group of species.
- Divide the set of species into two monophyletic groups:
  - The group of species displaying the shared novelty.
  - The complementary group of species, or single species.

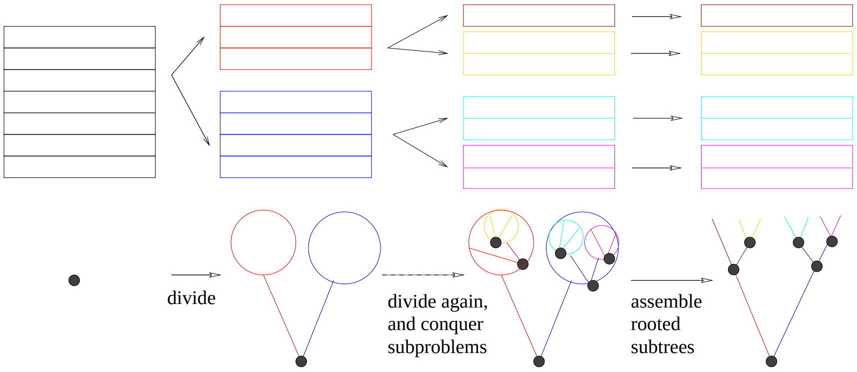
  Insert the corresponding vertex and appropriate edges into the tree.
- Call this procedure for each group recursively.

The complementary group is monophyletic because we use the shared novelty involving the *largest* group of species to define the split. Weak visibility is sufficient because for each recursion step we are dealing with a subset of the species for which we then have a (strongly) visible shared novelty.

The divide-and-conquer scheme is shown in Fig. 2.7. The top row displays the recursive split of an alignment of seven sequences, and the bottom row visualizes the corresponding knowledge that we have about the tree, established in a top-down manner.

The resulting tree and all subtrees are rooted, and for any kind of divide-and-conquer technique dividing sets of species, correct rooting of subtrees is a prerequisite for correct trees, since adding a differently rooted subtree introduces a different statement about the history of the species in that subtree, see Fig. 2.6, panels 3 and 4, on page 34. In panel 3, the subtree is attached to the remaining tree via the correct root between species 1-3 and 4,5. In panel 4, the subtree is rooted differently between species 1,2 and 3-5. Attaching it to the remaining tree results in a different overall tree.

Figure 2.7: The divide-and-conquer scheme.



2.8 Leftovers and Inconsistencies

In the last section, we have described a tree estimation method that works top-down, from the root to the leaves, finding the correct splits based on known visible shared novelties. However, usually these are not known. Therefore, we have to do heuristic searches for best-supported hypotheses of the correct split, encountering incorrect splits along the way, and the question is how to detect these. We will first deal with missplits, and then with unbacked splits.

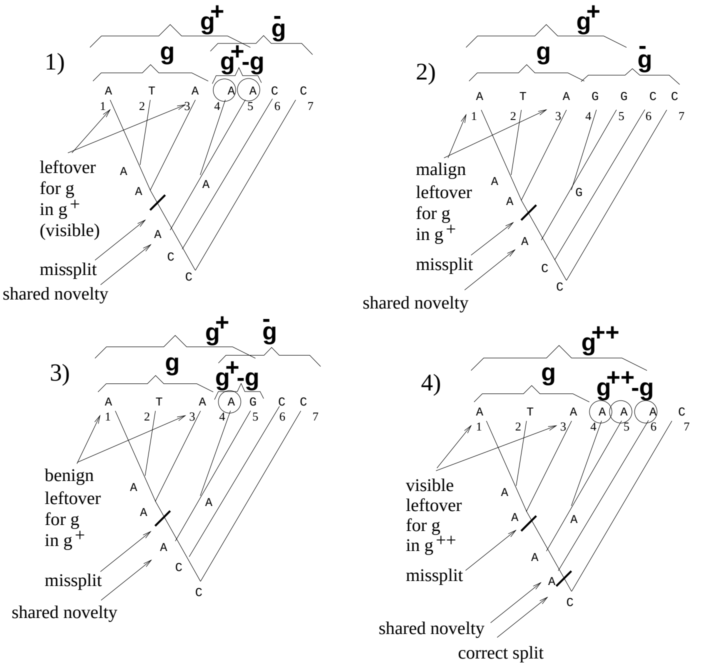
2.8.1 Benign Leftovers Show up in Missplits

A missplit $G = g$ v $\overline{g}$ (i.e. a split that still involves one monophyletic group g) can usually be detected by observing *shared old character states ("leftovers")* in $\overline{g}$, as follows.

Given a tree $\mathcal{T}$, a monophyletic group g of species, and a monophyletic supergroup $g^{+} \underset{\neq}{\supset} g$, a `shared leftover` for g *in* g^{+} is a shared novelty *in* g^{+} that was inherited by at least one species in g, cf. Fig 2.8, panel 1. A shared leftover, or `leftover` for short, is also called a `symplesiomorphy`, or a `shared old character state`. *It is an "old" state for g and defined with respect to g^{+},* in *which it is a new character state, a shared novelty that testifies its last common ancestor.*

It is important to note that up to now, all the "shared old character states" that we have dealt with in an informal way have been leftovers *in* the whole set of species under consideration, that is *in* $g \cup \overline{g}$. In other words, their point of reference is the whole set of species – they are shared novelties already featured by the last common ancestor of this set. That's why they cannot help to detect missplits in $g \cup \overline{g}$, since they are not shared novelties *in* a subset of $g \cup \overline{g}$ which could be torn apart by a missplit. (The whole set of species is *always* torn apart by a split – no information about the correct split is provided by shared

Figure 2.8: Visible, benign and malign leftovers.

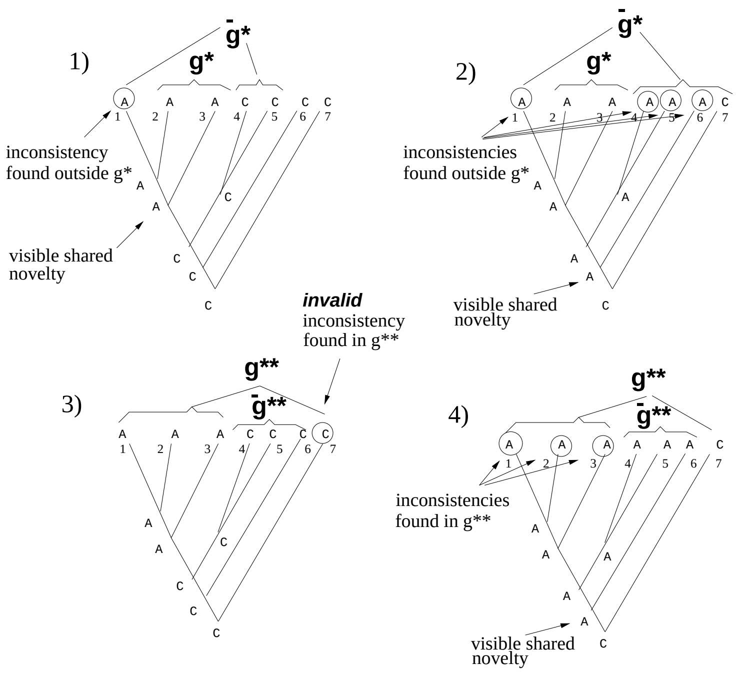


novelties *in* $g \cup \overline{g}$.) Shared old character states *in* the whole set of species can only misguide the analysis, if they are eroded away in some species.

Yet in other words, the value of leftovers depends on the group under consideration. They are useful only if they testify exclusive common heritage of a subgroup of the species under consideration, because they are shared novelties in these subgroups. Mathematically, the condition $g \cup \overline{g} \underset{\neq}{\supset} g^+$ ensures that g^+ is a subgroup of $g \cup \overline{g}$, and this ensures that the "leftover *in* g^+" need not be useless.

A `malign leftover` for g in g^+ is found in g only (Fig 2.8, panel 2), whereas a `benign leftover` for g in g^+ is found in the majority of species of g, and in $g^+ - g$, as in Fig 2.8, panel 3. A `visible leftover` for g in g^+ is a benign leftover present in *all* species of $g^+ - g$, cf. the first panel in Fig 2.8. Given no ancestral character states, benign leftovers can help us to see the conflict in a missplit. Only visible leftovers for g in g^+ can guide us towards the root of the tree, by identifying a larger monophyletic group g^+, as in panel 1. Only visible leftovers for g in g^{++} can identify the root, and therefore the correct split, $G^{++} = g^{++}$ v $\overline{g}^{++}$, cf. Fig. 2.8, panel 4. Note that *only the majority* of species of g must display a visible leftover. For example, it does not matter whether one species in g (like species 2 in all panels) features a character state

Figure 2.9: Valid and invalid inconsistencies.



that is different.

A benign leftover in g^+ may be confused with a shared novelty in g and a corresponding convergence if we don't know the ancestral character states.

A visible shared novelty in g^+ will trigger a visible leftover for any subgroup g of g^+. Invisible shared novelties may nevertheless give rise to visible leftovers; the conditions for this depend on the majority of species making up g. Some of the power of our method stems from the fact that visible leftovers do not require visible novelties, as in Fig. 2.8, panel 4.

2.8.2 Inconsistencies Flag Benign Leftovers

Given a split $G = g \text{ v } \overline{g}$, a character state in $\overline{g}$ that is part of a variable subcolumn is called an `inconsistency`, if it is matching with the majority character in g. Throughout this text, inconsistencies are marked by circles. Per definition, benign leftovers for g in g^+ are flagged by inconsistencies; these are found in $g^+ - g \subsetneq \overline{g}$. However, inconsistencies do not just flag leftovers. Inconsistencies also appear if we investigate unbacked splits, as shown in subsection 2.8.3, and they may be due to erosion and accumulation as discussed in subsection 2.8.4.

The term "inconsistency" is derived from the observation that an inconsistency violates a rule like "*only* the species in the monophyletic group g have character state `A`". In other words, the inconsistency reveals that the rule is inconsistent. On the other hand, substitutions of `A` within g testify incompleteness of the rule.

2.8.3 Inconsistencies Flag Apparent Leftovers in Unbacked Splits

In an unbacked split $G^* = g^* \text{ v } \overline{g}^*$, group g^* is not monophyletic, and we expect inconsistencies in the form of "apparent leftovers", cf. Fig 2.9, panel 1. Here, we investigate the assumption that g^* is monophyletic, but we observe an `apparent leftover` in $\overline{g}^*$ that features the same character state as g^* (`A` in the first species); g^* is the non-monophyletic subset of a monophylum and it does not include all species displaying the visible shared novelty. Fig. 2.9, panel 2 displays a similar situation. (Strictly speaking, we observe an apparent leftover in species 1, and a benign leftover in species 4-6.) In panel 3, group g^{**} is not monophyletic. However, the inconsistency in species 7 does *not* indicate this. It is *not* due to a shared novelty that testifies any monophyly within species 1-6. Instead, the inconsistencies in panel 4 are the apparent leftovers showing that g^{**} is not monophyletic.

2.8.4 Erosion, Accumulation and Leftovers

Finally, inconsistencies may also flag erosion and convergences. For example, the inconsistency in Fig. 2.9 panel 3 is due to erosion in species 1-3, and investigating the correct split 1-3 v 4,5 of species 1-5 in Fig. 2.6, panel 2, on page 34, the inconsistencies in species 5 are due to convergences. The inconsistencies in panel 1 of the same figure, where the correct split of species 1-5 is investigated as well, are the "`A`'s" found in species 1 and 3, and they are due to erosion in species 3.

We can say that our phylogeny estimation method will be based on penalizing leftovers, apparent or not. It tries to ignore inconsistencies due to erosion, and it may fail in the case of convergence accumulation, at least in the form presented. Furthermore, apparent and benign leftovers due to shared novelties may be overshadowed by inconsistencies due to erosion, if erosion affects exactly the species in a monophyletic group of the correct split. This is just rephrasing the observation we made for Fig. 2.4, panel 5, on page 32, where erosion happens in parallel with the evolutionary history of the species.

It is important that only character states in variable subcolumns are considered for inconsistency analysis; the species in $g \cup \overline{g}$ are assumed to be monophyletic anyway, and "inconsistencies" in fixed columns would just reveal this monophyly.

2.9 Inconsistency Patterns and Pattern Counts

Inconsistencies as defined in the last section are not relevant if they are just "scattered around" in the various species. What we are interested in are *patterns* of these.

Given a set of species $I = \{i_1, ..., i_\ell\} \subseteq \{1, ..., m\}$, a `pattern type` is simply a proper subset of I, excluding $\emptyset$ and I itself. Given an alignment A of length q and a split $G = g$ v $\overline{g}$, the two pattern types found in a character (alignment column) $j \in \{j_1, ..., j_q\}$ are

- $t = t(j)$, the list of species from g which display an inconsistency, and
- $\overline{t} = \overline{t}(j)$, the list of species from $\overline{g}$ which display an inconsistency.

Any or both lists may be empty. No duplicate entries occur in these lists such that we can always consider a canonical repeat-free list of species in numerical order, and we can also use set notation. In Fig. 2.8, panel 1, g is 1-3, and if $\overline{g}$ is 4-7, the pattern type in $\overline{g}$ is "4,5". In Fig. 2.9, panel 1, the inconsistency pattern type is "1". A `pattern` is the presence of a pattern type in one or more alignment columns. The split 1-3 v 4,5 in Fig. 2.6 on page 34 has pattern "1,3" in panel 1, and pattern "5" in panel 2, displayed in all 3 columns.

By inspecting the entire alignment, we can prepare two `lists of patterns`. For each subgroup g and $\overline{g}$, we list the subsets of species from the subgroup that display inconsistency patterns. The `columns supporting pattern` t in g are given by

$$\mathcal{C}(t) = \{j \in \{j_1, ..., j_q\}, \text{ such that for all } i \in \ g : a_{i,j} = c_j(\overline{g}) \iff i \in t\},$$

where

$$c(\overline{g}) = (c_{j_1}(\overline{g}), ..., c_{j_q}(\overline{g}))$$

is the majority sequence for $\overline{g}$. The columns supporting pattern t are the columns in which inconsistencies are observed exactly in the species making up t.

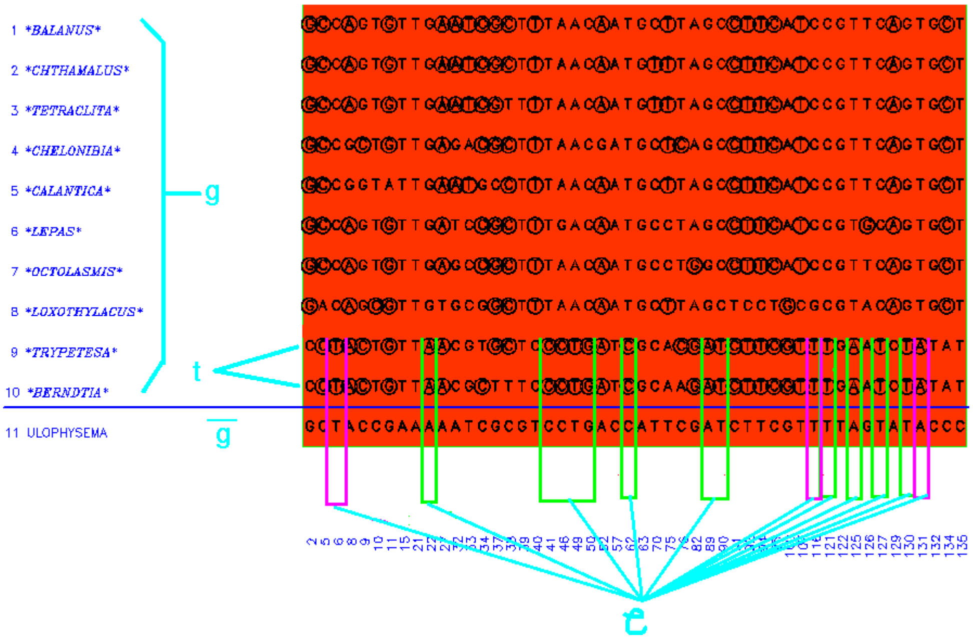
The `pattern count` of pattern t is the number of columns supporting t; it is also denoted by $s(t)$. The list of all patterns in g is $T = T(g) = \{t_1, ..., t_{|T|}\}$. $T(g)$ is empty if there are no inconsistencies in any subcolumn.

Running Example, Crustacean alignment

Let us continue our running example. In Fig. 2.10, we investigate the split $G = g$ v $\overline{g}$ = 1-10 v 11. The majority sequence of 11 coincides with the sequence itself. Its character states form a pattern t = "9,10" in columns $\mathcal{C}$ = 6, 22, 41, 46, 49, 50, 62, 89, 90, etc. (We will return to this figure very soon in the context of matching rates calculated by outgroup comparison.)

In Fig. 2.11, we investigate the split G = 1-8 v 9-11. (We have already investigated this split in Fig. 2.2.) In the first column, which is column 2 in the complete alignment of the dataset investigated in section 4.4.1, the majority

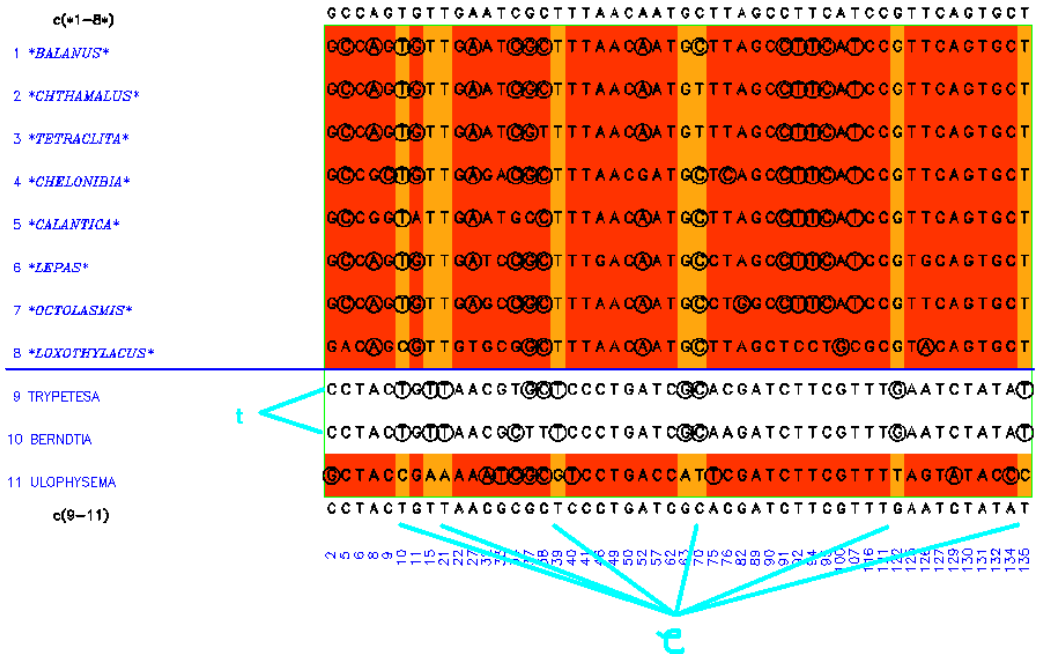
Figure 2.10: Inconsistency patterns in 1-10.



of 1-8 is G, and the majority of 9-11 is C. Species 11 displays an inconsistency, its character state matches the majority character state of the other group. This observation can be made for columns 2, 32, 33, 40, 75, 129 and 134. The character states of the majority sequence of 1-8 form another pattern ”9,10” in columns 10, 15, 21, 39, 63, 70, 122 and 135. The white background color flagging species 9,10 already hints at their better fit with species 1-8 than with species 11; it indicates a conflict due to the exclusive common heritage of species 1-10, and this conflict is found in the columns indicated. (The color coding of the alignment is based on a product of row scores and column scores. The ”species conflict” to be defined in section 3.9 is used as the row score, and the column score is defined analogously. Red indicates high scores, and white indicates low scores. The columns in question do not have a completely white background because there is no conflict in 1-8, yielding an average column score of one half.) We will return to this figure very soon.

In Fig. 2.12, we investigate the split G = 1-9,11 v 10. The majority sequence of 10 coincides with the sequence itself. Its character states display an inconsistency pattern t_1 = ”9” in columns $\mathcal{C}_1$ = 2, 32, 33, 40, 57, 75, 129, 132 and 134, and a pattern t_2 = ”1-9” in columns $\mathcal{C}_2$ = 15, 21, 39, 63, 122 and 135. Again,

Figure 2.11: Inconsistency patterns in 9-11.



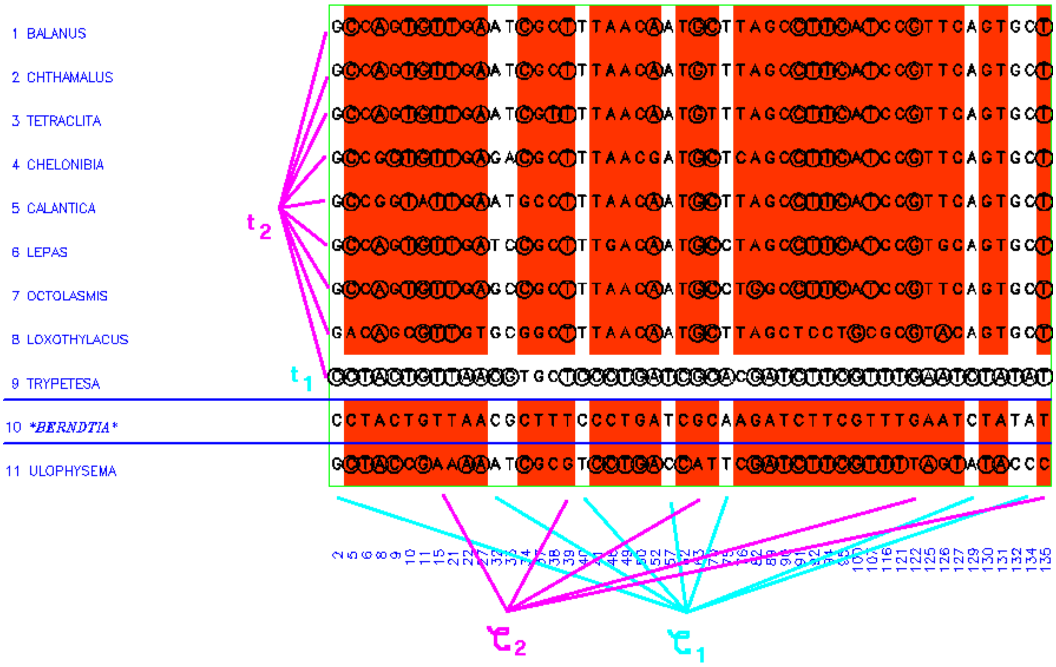
the white background for species 9 indicates that the split 1-9,11 is in conflict with the common heritage of species 9 and 10. (If we go beyond analyzing the first 50 columns, the other conflict, found in species 1-9 due to the common heritage of species 1-10, will become more pronounced.)

In Fig. 2.13, split G = 1-7,10,11 v 8,9 triggers 4 patterns, one in 8,9 and three in 1-7,10,11. The monophyly of 9,10 triggers inconsistencies in species 10 in 1-7,10,11. It does not trigger inconsistencies in species 9 in 8,9, except by coincidence, because species 10 cannot gain any majorities in 1-7,10,11. The monophyly of 1-8 triggers inconsistencies in species 8 in 8,9, and in species 1-7 in 1-7,10,11. Finally, the monophyly of 1-10 triggers inconsistencies in species 1-7,10.

2.10 Valid and Invalid Inconsistencies, and Divide-and-Conquer Revisited

We have just seen in section 2.8 that we can expect the observation of inconsistencies in incorrect splits. However, we need to distinguish valid and invalid

Figure 2.12: Inconsistency patterns in 1-9,11.



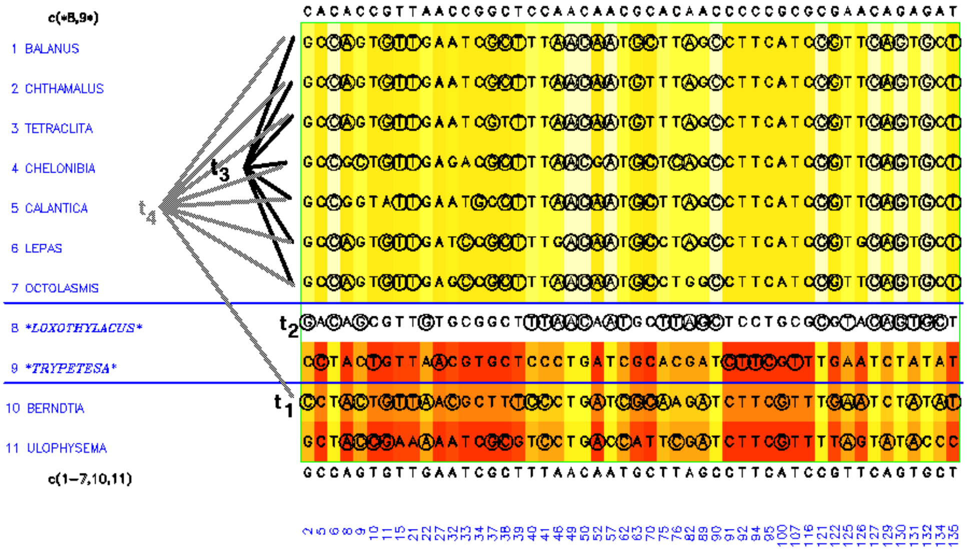
inconsistencies:

- In the case of a missplit, valid inconsistencies result from benign leftovers.
- In the case of an unbacked split, valid inconsistencies may flag apparent leftovers.
- In the case of *any* split, invalid inconsistencies may result from erosion and convergence accumulation.

In our running example, Fig. 2.10 gives an example for invalid inconsistencies, given the correct split. Fig. 2.11 and 2.12 give examples for valid inconsistencies triggered by a missplit, and Fig. 2.13 gives an example for valid inconsistencies triggered by an unbacked split.

Consequences resulting from the two types of inconsistencies are dramatically different: convergences and erosion must be ignored, while identification of valid inconsistencies implies a new hypothesis about monophyly. We will soon resort to a quantitative estimation of the phenomenon, that is a ”validity estimate” based in particular on the ”novelty estimate” calculated for an inconsistency pattern. It is designed to be proportional to the likelihood that shared

Figure 2.13: Inconsistency patterns in 1-7,10,11 and 8,9.



novelties torn apart or convergences are the reason for the inconsistency pattern. Of course, we would be much happier if our estimate were proportional to the likelihood that *only* shared novelties torn apart are the reason for the inconsistency pattern.

Basing invalidity of inconsistencies on erosion alone will nevertheless yield good empirical results, and we conjecture that erosion is the predominant phenomenon at least in the datasets we investigated. On the downside, we believe that one major failing condition of our method will occur if shared novelties are outvoted by convergence accumulation. For chance substitutions, we would expect that only one in three substituted nucleotides are convergences, but highly evolved species may nevertheless be subject to this problem. Convergence accumulation leads to conflict for the correct split, in the form of a seemingly valid inconsistency pattern. It is one of the reasons why we cannot always expect a minimum conflict close to zero. In particular, an increase in the frequency of some character states at the expense of others, like an abundance of A and T in some species only, may cause convergence accumulation. If, furthermore, there are incorrect splits that do not trigger conflict because there is no sufficient number of shared novelties that give rise to valid inconsistency patterns, our method fails because it favors an incorrect split that displays lowest conflict. If more than one split involves a conflict close to zero, at least we know that there is a problem. If, on the other hand, all splits involve some significant conflict, we also know that there is a problem. We can use multifurcating trees to highlight such problems – this is "Future Work".

As we have seen in Fig. 2.4, panel 5, on page 32, another failing condition may occur if erosion and the separation event leading to speciation occur in parallel. We will continue the discussion of these problems in section 3.5.2.

If we search for the split with minimum conflict, i.e. a minimum of valid inconsistencies, and then proceed top-down, from the root to the leaves, we obtain a heuristic version of the "ideal divide-and-conquer phylogeny estimation" presented in section 2.7 : Given a set of species G,

- Find the split with minimum conflict.
- Divide the set of species into two putative monophyletic groups, according to the split with minimum conflict.
  Insert the corresponding vertex and its edges into the tree.
- Call this procedure for each group recursively.

In section 3.13, we will give a detailed exposition of this algorithm.

Chapter 3

Minimum Conflict Phylogeny Estimation

3.1 The MCOPE Cascade

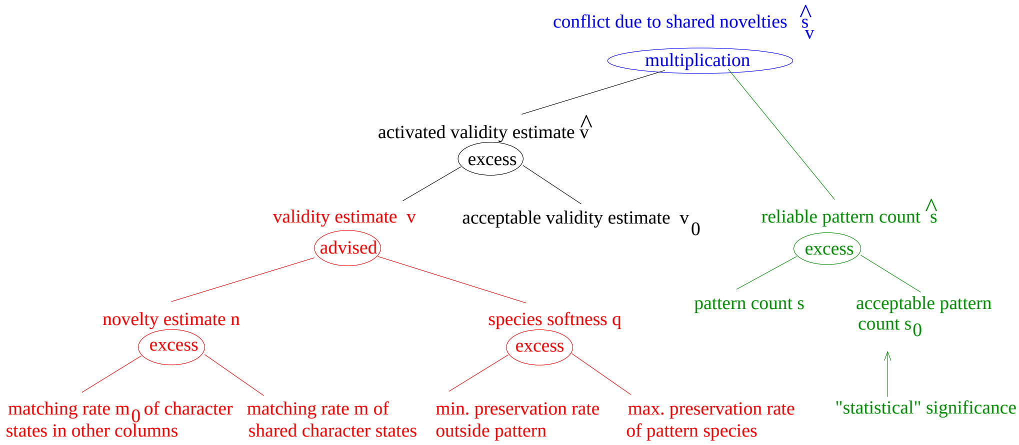
In the last chapter we have seen that the identification of inconsistency patterns, that is the systematic occurance of the majority character state of one group of species in another group of species, can be used to design a phylogeny estimation algorithm. The algorithm can be successful, if we are able to establish the *validity* of these patterns, that is the likelihood that they are due to shared novelties torn apart, and not due to branch-attraction artifacts that are caused by a different amount of evolutionary change in one of the groups. The following are the crucial steps of our approach:

- Based on outgroup comparison, we calculate an estimate for such a pattern validity.
- Then, we do a heuristic search for a minimum of conflict caused by "valid" patterns (see Fig. 1.3 on page 13).
- Finally, we apply our algorithm recursively (see Fig. 2.7 on page 36).

We will evaluate inconsistency patterns in a cascade of calculations designed to filter out invalid ones. These invalid patterns will include patterns with a *high* outgroup-based "validity estimate", *if* the number of supporting columns is so low that the validity estimate must be deemed unreliable. The cascade receives input from outgroup comparison via matching and preservation rates and from the actual number of columns supporting a pattern. Sigmoid functions are used to trigger clear decisions whenever possible.

Fig. 3.1 is a chart of the MCOPE cascade. Moving bottom-up in the chart, we will discuss the "novelty estimate" in section 3.5, and "species softness" in section 3.4. Both give rise to the "validity estimate" introduced in section 3.6. The "reliable pattern count" is discussed in section 3.7. The "conflict" due to shared novelties is the subject of section 3.8. First however, we will discuss sigmoid functions like *excess* and *advised*, which are an important generic ingredient of the cascade.

Figure 3.1: The MCOPE cascade.



conflict due to shared novelties $\hat{\check{s}}$
multiplication
activated validity estimate $\hat{v}$
excess
validity estimate v
acceptable validity estimate v_0
reliable pattern count $\hat{s}$
advised
excess
novelty estimate n
species softness q
pattern count s
acceptable pattern count s_0
excess
excess
matching rate m_0 of character states in other columns
matching rate m of shared character states
min. preservation rate outside pattern
max. preservation rate of pattern species
"statistical" significance

3.2 Sigmoid Functions

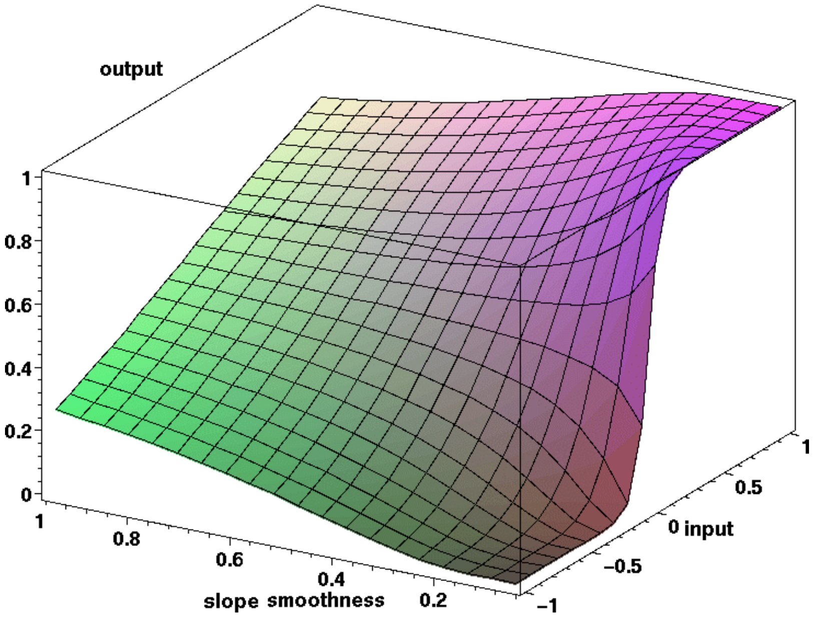
This section is concerned with ”sigmoid” functions that are used throughout the MCOPE cascade of signal amplification and filtering. These ”sigmoids” are applied whenever two values are compared, and their input is the difference of these two values. Their effect is twofold:

- The input is amplified whenever it is positive, and it is amplified the more, the larger it is.

- The input is squashed whenever it is negative, and it is squashed the more, the larger its absolute value is.

Both amplification and filtering (squashing) can be seen as a natural component of information processing, and the use of sigmoids is common in information processing applications from both engineering and computer science. (Some people believe that sigmoids are also a good model for some aspects of information processing in the brain, and if we interpret our method as a formalization of the application of Hennigian logic to molecular systematics by trained systematists, then we may conjecture that sigmoids model their decision process in some reasonable way.)

Figure 3.2: Sigmoid activation of a single input value.



The functions used are called "sigmoid" because they make, in one way or another, use of the sigma-shaped `standard activation function`

$$y = \frac{1}{1 + e^{-x/\theta}}$$

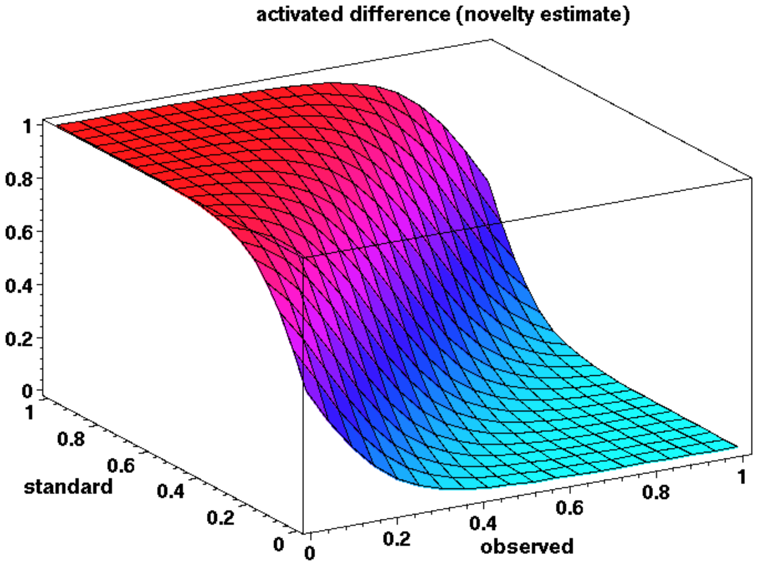
where $x \in]-\infty..\infty[$ is the input, and $\theta \in]0..\infty[$ is a scaling parameter controlling the smoothness of the slope. Fig. 3.2 displays the input-output correlation for different values of slope smoothness. The sigmoid activation function is also called the `logistic function`. (In some neural networks, the function is used to activate x, the weighted sum of neuron inputs, and the result is the neuron output).

We start with a treatment of "sigmoid" functions which

- activate the difference between two values,
- allow a value to advise another one,
- activate a count value, weighted by another activated variable, and
- implement a smooth OR-function.

Once we analyze the running time of our algorithm, we will assume that all these sigmoid functions take unit cost, i.e. $O(1)$.

Figure 3.3: Activation of the difference between two values.



3.2.1 Activation of Differences

For $\theta = 0.1$, Fig. 3.3 displays the activation of the difference of two values (*standard* and *observed*), with the property that the return value is *the higher, the larger the excess* of *standard* with respect to *observed* is. The formula is

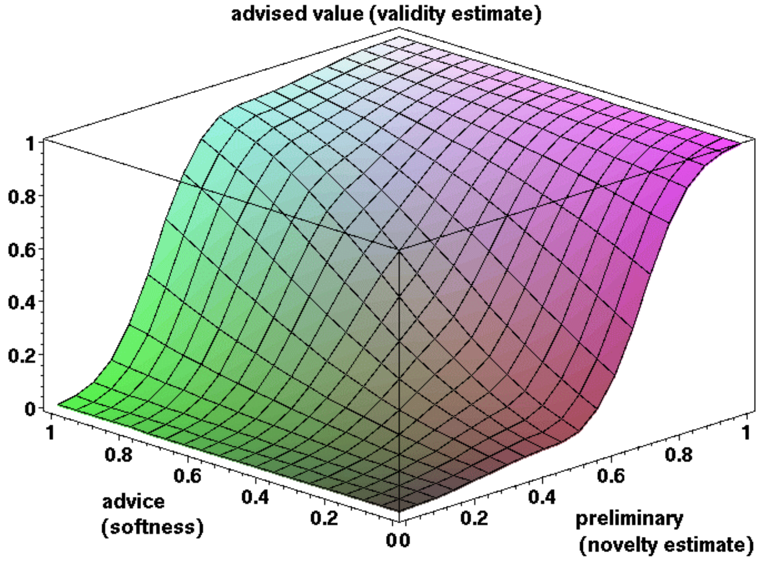
$$excess_{\theta}(standard, observed) = \frac{1}{1 + e^{-(standard - observed)/\theta}},$$

based directly on the logistic function. As an example, we will use this function to evaluate the difference between two outgroup matching rates, one (called the "observed matching rate") derived from the columns with shared character states and the other (called the "standard matching rate") derived from "the other columns". The comparison will yield higher "novelty" estimates the more the "standard" rate exceeds the "observed" one.

3.2.2 Advice

Based on a similar sigmoid formula, we can let some value *advisor*, $advisor \in [0..1]$, influence another value *preliminary*, $preliminary \in [0..1]$, depending on the ambiguity of the latter. See Fig. 3.4 for an example; the closer *preliminary* is to 0.5, the more advice is taken. The underlying formulas are, given a smooth-

Figure 3.4: The "advice" value influences the "preliminary" value.



ness parameter θ_n,

$$\omega = abs(preliminary - 0.5) + 0.5,$$

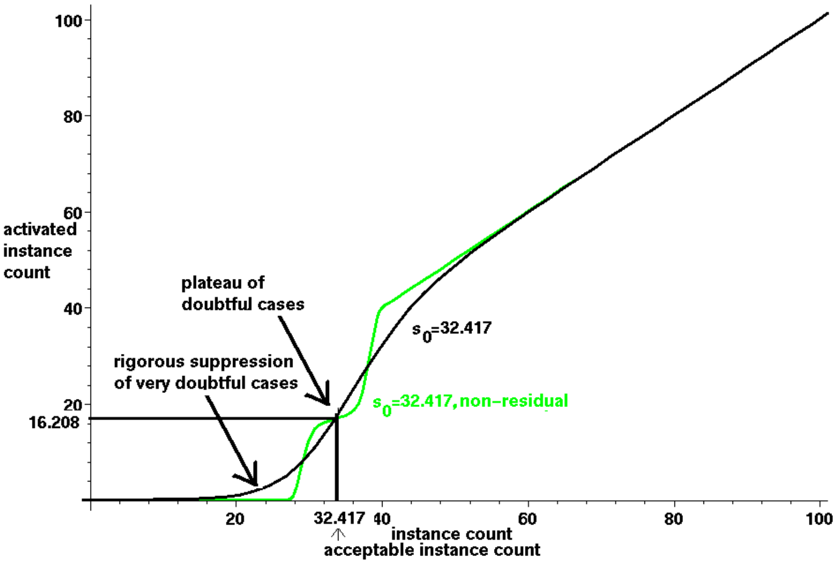
$$\Sigma = \omega \cdot preliminary + (1 - \omega) \cdot advisor,$$

$$advised_{\theta_n}(preliminary, advisor) = \frac{1}{1 + e^{-(\Sigma - 0.5)/\theta_n}},$$

where ω is the weight that is given to *preliminary* depending on its ambiguity, Σ is the weighted sum of both *preliminary* and *advisor*, and the final result is an activation of this weighted sum. If *preliminary* is 1, the weight ω is 1, and the weighted sum Σ is 1 as well, and if *preliminary* is 0, ω is 1, and Σ is 0 as expected. For other values of *preliminary*, the weight is at least 0.5, and the weighted sum is between 0 and 1. To match this [0..1] interval to the input interval of the standard sigmoid function, we need to transform the weighted sum by substracting 0.5, resulting in an interval with 0 in the middle.

In Fig. 3.4, the *advised* function is shown, where θ_n is set to the standard 0.1. We will use this function to give some influence on the novelty estimate to a weaker criterion of novelty, i.e. the "species softness", yielding the so-called "validity estimate".

Figure 3.5: Activation of an instance count via different formulas.



3.2.3 Amount of Evidence

The values activated up to now are in the interval $[-1..1]$, but sigmoid activation works as well for values in the interval $]-\infty..\infty[$. In particular, we can compare two instance count variables, and we can weight an instance count variable s with the result of its comparison to a threshold, s_0:

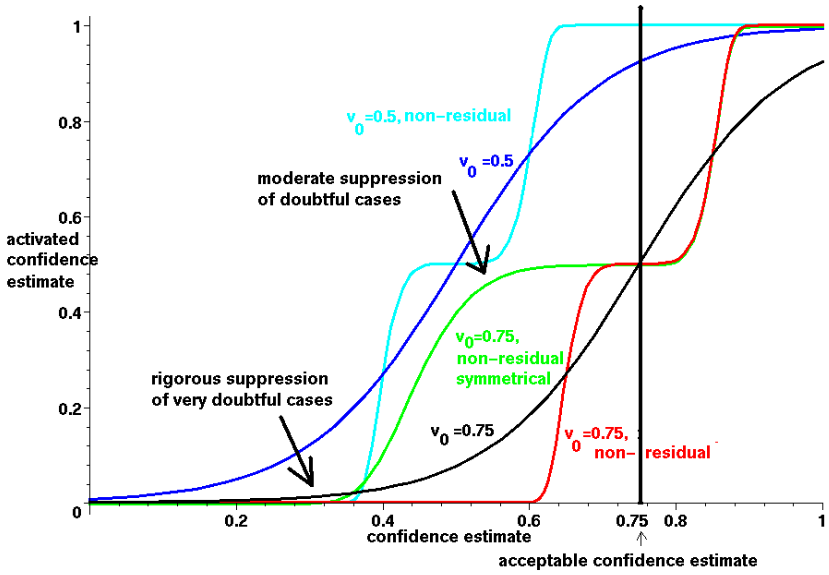
$$excess_{\theta_s}(s, s_0) \cdot s = \frac{1}{1 + e^{-(s-s_0)/\theta_s}} \cdot s.$$

The more s exceeds s_0, the more s can retain its value. Given s, $excess_{\theta_s}(s, s_0) \cdot s$ will be interpreted as the number of instances that we have available to reach *reliable* conclusions based on some property of these.

If we observe an instance pattern count $s \in]0..\infty[$, and are given an `acceptable instance count` s_0, to which we assign a reliability of one half, the preceding formula will result in one half the observed count, if the observed is equal to the acceptable count, i.e. $s = s_0$. The formula will yield activated counts close to zero if the observed count is far below the acceptable one, $s << s_0$, but it will not suppress an observed count that is far ahead, $s >> s_0$.

The sigmoid activation of instance counts is shown in Fig. 3.5, in black (for $s_0 = 32.417$). We will soon discuss the other plot in this figure – we will prefer such a curious kind of activation for reasons that will be explained soon. The

Figure 3.6: Activation of a confidence estimate via different formulas.



odd number 32.417 is taken from the sample data of section 4.4.1, calculated as explained in section 3.7 starting on page 80.

Sigmoid activation may also be used to activate the difference between an observed confidence estimate, $v \in [0..1]$, and a standard, `acceptable confidence estimate`, $v_0 \in [0..1]$:

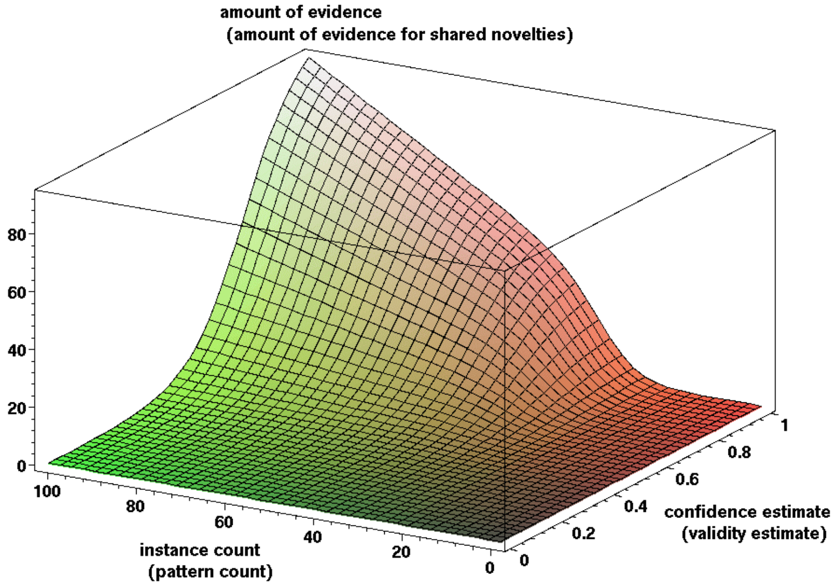
$$excess_{\theta_v}(v, v_0) = \frac{1}{1 + e^{-(v-v_0)/\theta_v}}.$$

Given v, $excess_{\theta_v}(v, v_0)$ will be interpreted as the amount of confidence that we have in some property under investigation. Strong confidence is amplified, weak confidence is squashed.

The slope smoothness θ_v is set to the usual 0.1. The sigmoid activation of the confidence estimate is shown in Fig. 3.6, in blue (for $v_0 = 0.5$), and in black (for $v_0 = 0.75$). We will soon discuss the other plots in this figure – suffice it to say that we will settle with the green plot. (Care has been taken to render the color designations redundant wherever necessary, by labelling all plots in a way that identifies them directly).

Combining both formulas, we will now design an activation scheme that takes an instance count and a corresponding confidence estimate, and thresholds (acceptable values) for each. It suppresses small instance counts as well as any instance count with low confidence, by *multiplying* the activated instance count

Figure 3.7: Weighted activation of instance counts .



by its activated confidence estimate, and it yields the `amount of evidence`.

For example, the input can be a pattern count and its validity estimate. Then we suppress patterns with small counts and/or low validity, and we obtain what we will call the "amount of evidence for shared novelties" that is behind the conflict in an inconsistency pattern.

As another example, imagine that we listen to a piece of music via short-wave radio. Due to distortion, we can just listen to low-quality fragments. What is the amount of evidence that we listen to a piece of music known to us ? At least two conditions must be fulfilled:

- We cannot tell from just a few fragments even if we recognize them well; the similarity may be due to chance alone.
- If we do not recognize the fragments (low confidence estimate), there is no evidence for a known piece either, no matter how many fragments we hear.

The formula for the amount of evidence is as follows.

$$evidence(s, v) = excess_{\theta_s}(s, s_0) \cdot s \cdot excess_{\theta_v}(v, v_0).$$

See Fig. 3.7 for a display of this function in case of $s_0 = 32.417$, $v_0 = 0.75$, $\theta_s = 4.322$ and $\theta_v = 0.1$.

In general, both s_0 and θ_s will be estimated from the data as explained in section 3.7; the odd values taken here are the ones derived from the example data discussed in section 4.4.1. An acceptable confidence estimate of $v_0 = 0.75$ follows from the idea that the activated confidence estimate is used as the multiplier for the activated instance count. Then, if the confidence estimate is 0.75, right in the middle between 0.5, which is the case of doubt, and the maximum at 1, we want the multiplier to be 0.5, cutting the activated instance count by one half. As before, a slope smoothness $\theta_v = 0.1$ is standard for values activated in the interval $[-1..1]$.

Using the plain formula for $\mathit{evidence}(s, v)$, a small confidence estimate will still trigger a *non-zero* activated confidence estimate. Multiplying a sufficiently large activated instance count by such non-zero activated confidence will incorrectly flag some residual amount of evidence. In the "piece of music" example, lots of fragments that we do not deem recognizable will nevertheless contribute some evidence, *if we are not completely sure,* that is, *if our confidence is slightly larger than zero.*

In our case, multiplying activated inconsistency pattern counts by their activated validity estimate, patterns with a very small validity estimate can still flag shared novelties if their count is high enough. Let us assume that a pattern is reliable, but due to erosion. It occurs in 250 columns, and its validity estimate is 0.25. Then we have,

$$\mathit{evidence}(250, 0.25) = 250 \cdot \mathit{excess}_{0.1}(0.25, 0.75) = 250 \cdot 0.0067 = 1.673.$$

Even a small validity estimate of 0.1 still flags a small residual amount of evidence for shared novelties,

$$\mathit{evidence}(250, 0.1) = 250 \cdot \mathit{excess}_{0.1}(0.1, 0.75) = 250 \cdot 0.0015 = 0.375,$$

which most certainly is an artifact.

To achieve a `non-residual activation`, we simply multiply the scaled difference ((0.25-0.75)/0.1 = -5) with itself a number of times:

$$(-5)^3 = -125, \ (-5)^5 = -3125.$$

For the examples just discussed we obtain, using exponent 5,

$$\mathit{evidence}_5(250, 0.25) = 250 \cdot \frac{1}{1 + e^{-((0.25-0.75)/0.1)^5}} = 250 \cdot 0 = 0$$

and

$$\mathit{evidence}_5(250, 0.1) = 250 \cdot \frac{1}{1 + e^{-((0.1-0.75)/0.1)^5}} = 250 \cdot 0 = 0.$$

Since scaled differences may be negative, valid exponents are the odd numbers 3, 5, 7, etc. The modified formula for the non-residual activation of a confidence estimate v is

$$\mathit{excess}_{\theta_v,\eta}(v, v_0) = \frac{1}{1 + e^{-((v-v_0)/\theta_v)^\eta}}.$$

Analogous arguments can be brought up in favor of a non-residual activation of an instance count s, and the corresponding formula is

$$excess_{\theta_s,\eta}(s, s_0) = \frac{1}{1 + e^{-((s-s_0)/\theta_s)^\eta}}.$$

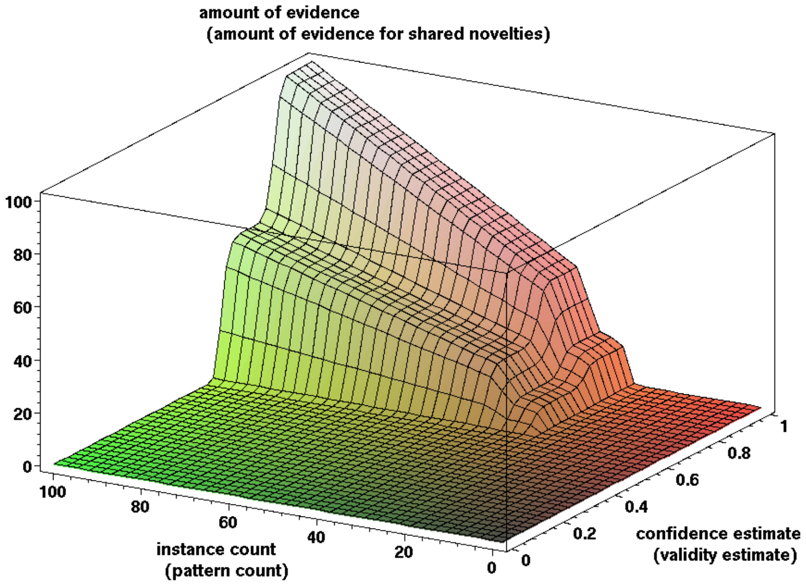
In both formulas η can be any odd integer.

The combined formula is

$$evidence_\eta(s, v) = excess_{\theta_s,\eta}(s, s_0) \cdot s \cdot excess_{\theta_v,\eta}(v, v_0).$$

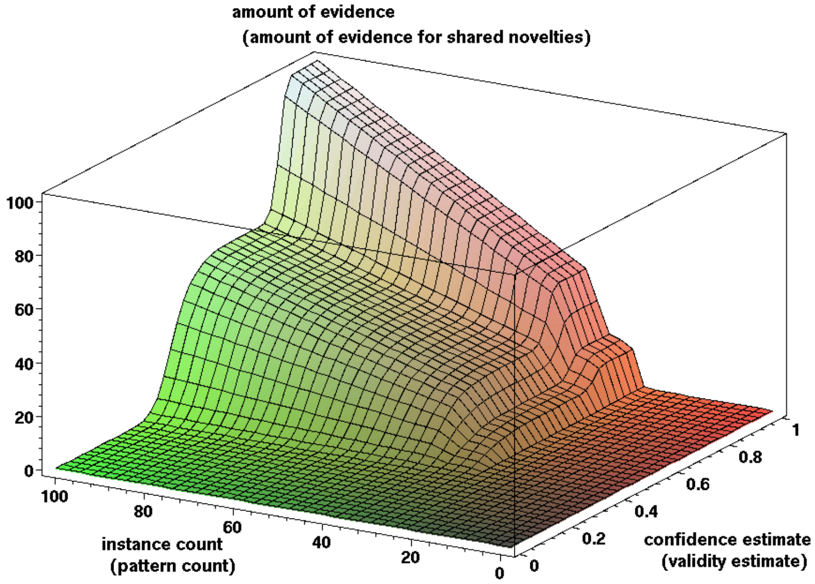
A plot of this formula is displayed in Fig. 3.8 using the same parameters as in Fig. 3.7, and $\eta = 5$. The non-residual activation of a count variable is given in Fig. 3.5, in green ($s_0 = 32.417$, non-residual) and the same activation of the confidence estimate is shown in Fig. 3.6, in red ($v_0 = 0.75$, non-residual).

Figure 3.8: Non-residual weighted activation of instance counts .



A second issue arises if the acceptable confidence estimate is not 0.5. Then, the intervals to the left and to the right of the threshold value are of different length, even though the smoothness of the slope is the same. For example, using an acceptable confidence estimate of 0.75, confidence estimates are completely suppressed as they approach 0.5, as can be seen for the red plot ($v_0 = 0.75$, non-residual) in Fig. 3.6. If we wish to give influence to "cases of doubt" with confidence estimates around 0.5, we may use a "translation" and move the

Figure 3.9: Symmetric, non-residual weighted activation of instance counts .



acceptable confidence estimate towards 0.5, cf. the cyan plot ($v_0 = 0.5$, non-residual) in Fig. 3.6. Or, we may use `symmetric scaling`. Symmetric scaling simply means that all scaled differences stemming from the larger subinterval, like $[0, 0.75[$, are divided by 3, because the interval is 3 times larger than the smaller subinterval $[0.75, 1]$. More formally, given an acceptable confidence estimate v_0, we rescale the interval to the left, dividing by $\frac{v_0}{1-v_0}$, which is the ratio of the lengths of the two intervals. For $v_0 = 0.75$, we obtain a ratio of 0.75:0.25, or 3:1.

In the "piece of music" example, we now maintain that sufficiently many fragments can give evidence even if we are in doubt. In our case, a pattern with doubtful validity will flag some amount of evidence for shared novelties if it occurs with sufficient frequency. In both cases, very strong doubt will still give *no* residue of evidence.

Symmetric scaling can be expressed by the following formula:

$$excess^{\bowtie}_{\theta_v,\eta}(v, v_0) = \begin{cases} \frac{1}{1+e^{-((v-v_0)/\theta_v)^\eta}} & \text{if } v \geq v_0, \\ \frac{1}{1+e^{-((v-v_0)/(\frac{v_0}{1-v_0}\cdot\theta_v))^\eta}} & \text{otherwise .} \end{cases}$$

A plot of the combined formula is displayed in Fig. 3.9, again using the same parameters as in Fig. 3.7. The underlying formula is now

$$evidence^{\bowtie}_{\eta}(s, v) = excess_{\theta_s,\eta}(s, s_0) \cdot s \cdot excess^{\bowtie}_{\theta_v,\eta}(v, v_0),$$

adding the $\bowtie$ symbol and the exponents η to the formula on page 53. For the axis of the confidence estimate, the smoothness of the slope is now in line with the size of the interval. The symmetric non-residual activation of the confidence estimate is shown in Fig. 3.6, in green ($v_0 = 0.75$, non-residual, symmetrical). Although both symmetric scaling and translation give more weight to cases of doubt, they yield very different results in practice, because in case of symmetric scaling, the threshold yielding an activated confidence estimate of 0.5 can still be 0.75, even though estimates below 0.5 are not completely suppressed.

In summary, we want to ensure that the following two conditions apply to the amount of evidence:

- Suppress doubtful cases ($1/3 \leq v < 3/4$) moderately by more than one half, but don't ignore these.
- Suppress very doubtful cases ($v < 1/3$) rigorously, even if the instance count is very high.

We conjecture that strong symmetric scaling is the natural way of accomplishing this. As can be seen from Fig. 3.6, the first condition is not met for $v_0 = 0.5$, irrespective of non-residual activation (blue and cyan plots), because the suppression is insufficient. The first condition is not met either for $v_0 = 0.75$ with non-residual activation (red plot), because the suppression is too strong. The second condition cannot be met without non-residual activation because otherwise suppression is insufficient for $v < 1/3$ (black plot, $v_0 = 0.75$). The green plot ($v_0 = 0.75$, non-residual, symmetrical) is the only survivor; it meets both conditions.

3.2.4 Smooth OR

Following up on the formula for advice presented in subsection 3.2.2, we can design a smooth way to take the "OR" of two values in the interval $[0, 1]$, by just weighting both values independently, as follows.

Given a smoothness parameter θ_{hq}, we set

$$\omega_{x_0} = abs(x_0 - 0.5) + 0.5,$$

$$\omega_{x_1} = abs(x_1 - 0.5) + 0.5,$$

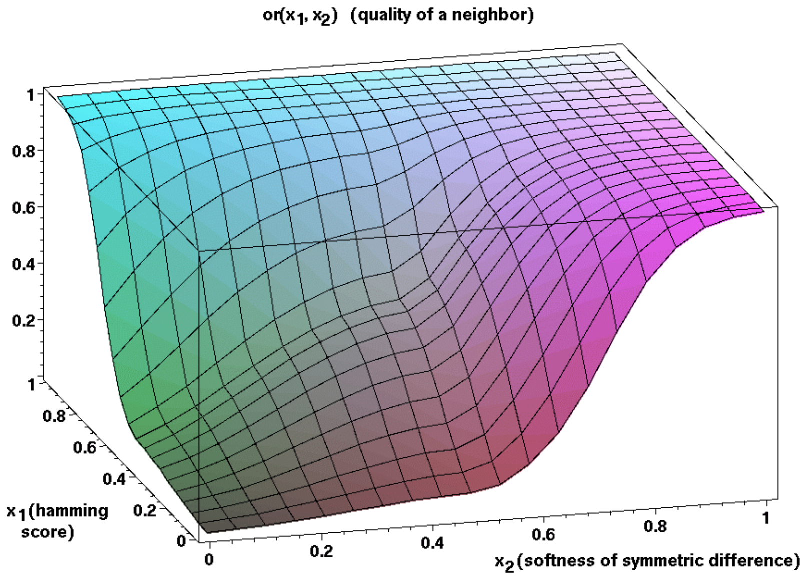
$$\Sigma = \omega_{x_0} \cdot x_0 + \omega_{x_1} \cdot x_1,$$

where ω_{x_0} is the weight that is given to x_0, ω_{x_1} is the weight of x_1, and Σ is the weighted sum of the input values. Their smooth OR evaluates to

$$or_{\theta_{hq}}(x_0, x_1) = \frac{1}{1 + e^{-(\Sigma - 0.5)/\theta_{hq}}}.$$

A plot of this function is shown in Fig. 3.10, for the standard slope smoothness $\theta_{hq} = 0.1$. Given an inconsistency pattern, we will use smooth OR to estimate the possibility that another pattern is just its transformation, and that both should be treated alike. This possibility depends on the hamming distance of the two patterns, and the "softness" of their symmetric difference. Basically, if only one of these two criteria is met clearly, the possibility is high, and if both criteria are met moderately, it is high as well.

Figure 3.10: Smooth OR.



It is a nice observation that empirically, a good slope-smoothness parameter for any activation of values in the interval $[0,1]$ is 0.1. A thorough, but not exhaustive investigation indicates that 0.1 is also a close-to-optimal value for the biological data that we will deal with. We will even use the ratio of $\theta_v = 0.1$ to $v_0 = 0.75$ as the criterion to calculate θ_s, given s_0. (And indeed, $0.1/0.75 = 4.322/32.417$, cf. page 53).

3.3 Matching Rates and Preservation Rates

We will now turn our attention to the identification of substitutions in the alignment of sequences belonging to a set of species. The preservation rates defined in this analysis will serve several purposes:

- They give rise to a weak estimate of pattern novelty which we will call "species softness".
- They are used to guide outgroup maintenance.

- They play a role in calculating a similarity score on the set of pattern types.

The first two issues are discussed in section 3.4, purpose number 3 is highlighted in section 3.10, and the final item is followed up in section 3.12. Moreover, in this section we will define matching rates in general. Specific matching rates will play a major role in the upcoming sections.

Tackling a split $G = g \text{ v } \overline{g}$, we cannot precisely identify substitutions (and, subsequently, erosion or convergence accumulation) unless we know the ancestral sequences. However, a rough estimate is possible if an outgroup gO is given. An `outgroup` gO for $G = g \text{ v } \overline{g}$ is a set of species that are not among the descendants of the last common ancestor of G, $lca(G)$. Nevertheless, the majority sequence of the outgroup is used as an estimate for $lca(G)$. Our method assumes that this is a good estimate for purposes of matching rate *comparison*. (We will assume that the outgroup is of constant size, or of size linear in the number ℓ of species currently studied.)

A `matching rate` is the frequency of matching character states, defined for two disjoint groups of species that are to be compared, and a set of columns J in which the comparison takes place. In formal terms,

$$m(I_1, I_2, J) = \frac{|\{j \in J : c_j(I_1) = c_j(I_2)\}|}{|J|},$$

where I_1 and I_2 are the two disjoint groups of species, and $c_j(I)$ is the majority character state of the species making up I, at column j.

Given two matching rates $m(I, I^*, J_1)$ and $m(I, I^*, J_2)$, and sufficiently large column sets J_1 and J_2, we will assume that a change of I^* to I^{**} does not have a major effect on their quotient. We assume that both matching rates go up if group I^{**} is closer to I than I^* is, and that they both go down if group I^{**} is farther away.

More specifically, given a set of species t, where $t \subset g$ or $t \subset \overline{g}$, an outgroup gO and two sets of columns J_1 and J_2, we evaluate quotients of the form

$$\frac{m(t, gO, J_1)}{m(t, gO, J_2)}.$$

If we assume that for any two groups of species I^* and I^{**}, which do not need to be recent, and for all sufficiently large column sets J,

$$m(t, I^*, J) \approx \pi(I^*, I^{**}) \cdot m(t, I^{**}, J),$$

where π is a constant depending of the length of the path between I^* and I^{**}, then we have approximate equality of quotients as follows:

$$\frac{m(t, gO, J_1)}{m(t, gO, J_2)} \approx \frac{m(t, lca(g), J_1)}{m(t, lca(g), J_2)},$$

for a wide range of outgroups gO and sufficiently large column sets J_1 and J_2. In other words, the outgroup is a good estimate for the last common ancestor of g for the purpose of matching rate comparison.

So let us assume that we are able to provide an outgroup for each split of species we tackle. In the beginning, the outgroup is supplied by the user. Outgroup maintenance across the divide-and-conquer steps is discussed in section 3.10.

A `deviation` in G is a character state in a species $i \in G$ which is not matching with the one estimated for the last common ancestor of G. If the estimation is correct, and if no substitutions back to the "ancestral" state have occured, a deviation is just a substitution, and vice versa, cf. Fig 2.3, panel 3 on page 30, where the ancestral character states are known. Another example are the nucleotides marked by squares in Fig. 3.11, where the outgroup is used as an estimate for the ancestral states. A character state that is not a deviation is called a `"preserved"` character state. (The quotation marks remind us that a deviation does not need to be a substitution after all.)

Let an alignment $A = (a_i)_{i \in \{i_1,...,i_\ell\}} = a_{i_1}, ..., a_i, ..., a_{i_\ell}$, where $\{i_1, ..., i_\ell\} \subseteq \{1, ..., m\}$, of length q be given. As usual, we ignore fixed columns. We can calculate a preservation rate for individual species by calculating the relative number of "preserved" character states displayed. Given an outgroup gO, the `preservation rate` of species i is defined as

$$p(i) = m(i, gO, \{j_1, ..., j_q\}) = \frac{|\{j \in \{j_1, ..., j_q\} : a_{i,j} = c_j(gO)\}|}{q},$$

where

$$c(gO) = (c_{j_1}(gO), ..., c_{j_q}(gO))$$

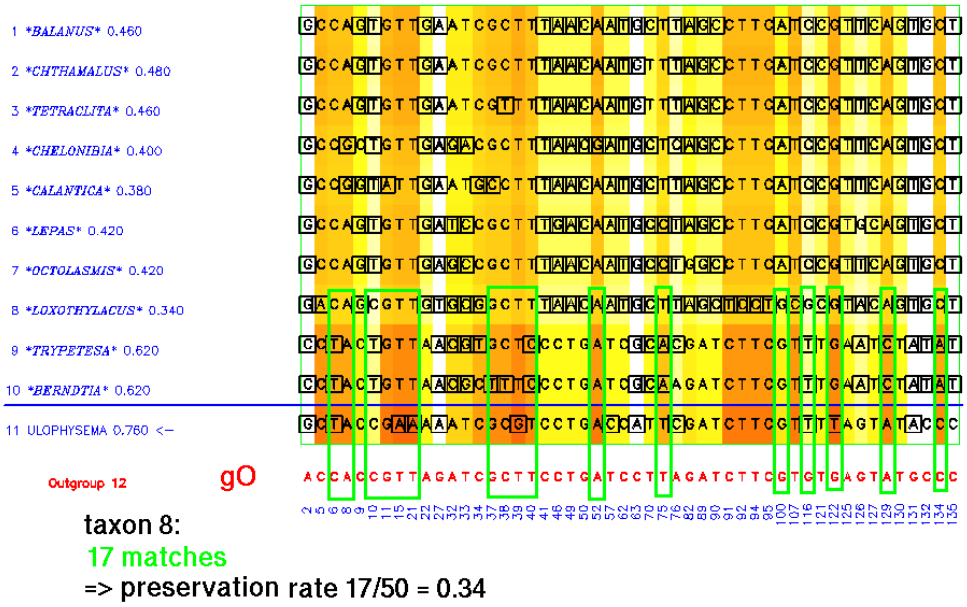
is the outgroup majority sequence. Note that $c_j(i)$ is $a_{i,j}$. In the next section, we will start comparing matching rates in the form of preservation rates calculated for various species.

The time complexity for calculating the preservation rate of all species in g is $O(\ell q)$, since the outgroup majority sequence takes at most $O(\ell q)$ for an outgroup of size $O(\ell)$, and preservation rate calculation takes at most $O(q)$ per species.

Running Example, Crustacean alignment

In Fig. 3.11, we consider the same alignment and the same split as in Fig. 2.10 on page 41, but marks and color codes are for preservation analysis, not inconsistency analysis. In fact, the split does not matter for preservation analysis; we could have taken any other one. The outgroup (species 12) is printed in red. Squares are used to mark deviations, and the preservation scores are listed next to the species names. There are 50 variable columns. The preservation of the first species, Balanus, then is $0.460 = \frac{50-27}{50}$, since there are 27 deviations from the outgroup. We note that preservation in species 8 is particularly low, and species 11 is the most preserved. The color coding of the alignment is simply the product of the preservation rate of the species and an analogous column score; high values are indicated by red, low values trigger white color. As expected, species 8 stands out, as do columns 2, 27, 63, 131, 132 and 135. (As before,

Figure 3.11: Preservation rates in 1-11.



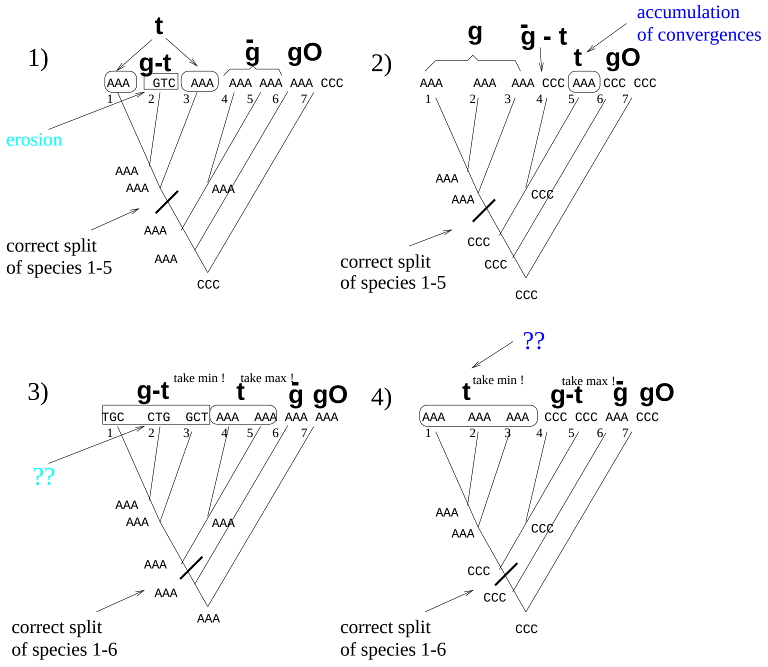
column numbering is done with respect to the original alignment including sites that are fixed, have gaps, etc.)

3.4 A Weak Estimate for Pattern Novelty

The preservation rates assigned to individual species in section 3.3 can easily serve to calculate a weak estimate for novelty, as follows. For all calculations, we assume that suitable outgroups are given, see also section 3.10.

Remember the concepts of erosion and convergence accumulation introduced in section 2.6, Fig. 2.4, panels 3 and 4, on page 32 and in Fig. 2.6, panels 1 and 2, on page 34. Displayed in a tree, these are investigated again in Fig. 3.12, panels 1 and 2. Let's ignore species 6 and 7 for the moment, and investigate species 1-5. Due to erosion, substitutions in species 2 trigger the illusion of shared novelties in species 1,3-5. This has consequences for the split 1-3 v 4,5, which is the correct split if species 1-5 are under investigation: We observe pattern 1,3 in 1-3 v 4,5. If we also observe a low preservation rate in species 2

Figure 3.12: Interpreting inconsistency patterns, I.



(which should not just be based on the columns displayed in our tree), we have evidence that erosion is yielding the illusion of shared novelties in species 1,3. The pattern may be invalid.

In panel 2, let's again investigate 1-5 and ignore species 6 and 7. Low preservation in species 5 (which should not just be based on the columns displayed in our tree) gives evidence that we are victim of an illusion of shared novelties in 1-3,5; a case of convergence accumulation. In other words, we observe an invalid pattern ”5” in group 4,5 of the correct split 1-3 v 4,5.

Consider Fig. 3.12, panel 3. Now the subject of our investigation is group 1-6, and not group 1-5 as in panel 1. Also, some character states have been changed. We change perspective because we will be talking about maxima and minima, and this is more instructive if there are at least two values from which these are taken. This would not be the case in panel 1.

Investigating species 1-6, the correct split 1-5 v 6 triggers the pattern $t = 4{,}5$ in 1-5. An indicator for *lack* of erosion would be that the preservation rate in species 1-3 is larger than the rate in the other species of 1-5, i.e. 4,5. (This is not the case for the three columns shown, but it may still be the case considering other columns!) More precisely, *lack* of erosion is indicated if the minimum preservation rate in $g - t$ is in excess of the maximum preservation rate in t:

$$\min_{i \in g-t} p(i) > \max_{i \in t} p(i).$$

The *minimum* preservation rate in $g - t$ is taken since an inconsistency pattern t cannot develop by erosion if the least preserved species in $g - t$ is still very preserved. Then, many old character states are preserved in *all* species in $g - t$. (For an inconsistency pattern to exemplify erosion, the converse needs to hold true: The old character states do not tend to be preserved in *any* species in $g - t$.) The *maximum* preservation rate in t is taken since an inconsistency pattern t cannot develop by erosion if the most preserved species involved is still very *un*preserved. Then, the old character states are not preserved in *any* species in t. (For an inconsistency pattern to exemplify erosion, the converse needs to hold true: The old character states tend to be preserved in *all* species in t.)

To quantify the excess, we use a sigmoid function as explained in section 3.2.1. The `species softness` q of the species involved in pattern t is given by the expression

$$q = q(t) = excess_{\theta_p}(\min_{i \in g-t} p(i), \max_{i \in t} p(i)).$$

The slope-smoothness for this activation is θ_p, which is set to the standard 0.1. In Fig. 3.13, a plot of this function is shown. Apart from the labels, it is identical to Fig. 3.3 on page 49.

If the species involved in pattern t are soft, we have a weak hint that *no* erosion took place. Again, the reader should note that preservation rates are estimated along the whole sequence with respect to the outgroup species gO, and do *not* need to refer to the sites displayed in Fig. 3.12.

In panel 4, we investigate accumulation, following up on panel 2. We change the subject of our investigation to group 1-6, like we did for panel 3, and we modified some character states, too. Furthermore, the pattern now appears in group g, and not in group $\overline{g}$. For this pattern type t = 1-3 in group 1-5, an indicator of *lack* of convergence accumulation is that the preservation rate in species 1-3 is larger than the rate in the other species in 1-5, i.e. 4,5. (Again, this is not the case for the three columns shown!) More precisely, lack of accumulation is indicated if the minimum preservation rate in t is in excess of the maximum preservation rate in $g - t$:

$$\min_{i \in t} p(i) > \max_{i \in g-t} p(i).$$

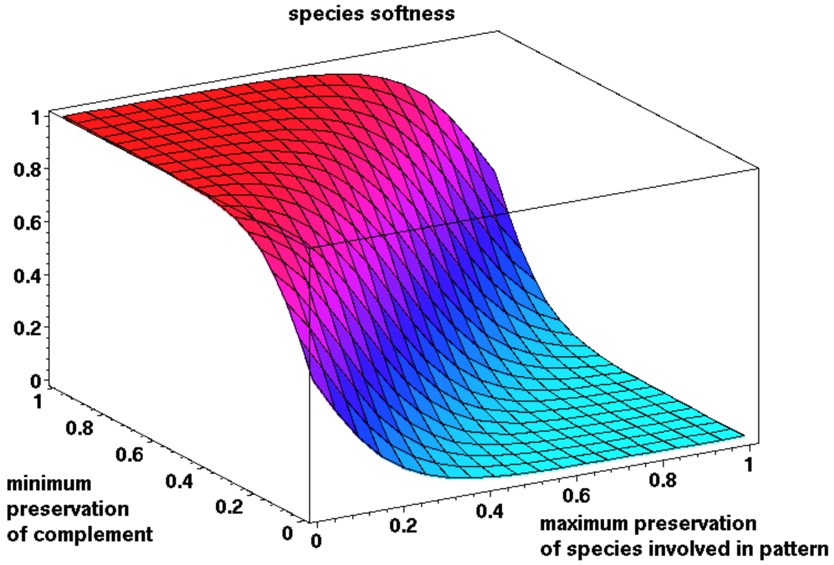
The `species resistance` is given by the expression

$$r = r(t) = excess_{\theta_p}(\min_{i \in t} p(i), \max_{i \in g-t} p(i)) = q(g-t).$$

If one group is soft, the other is resistant, and vice versa.

At the time of this writing, species resistance is a concept that we fail to include into the overall approach, because softness and resistance estimates tend

Figure 3.13: Species softness



to be inversely correlated. If $|g| = 2$, minima and maxima coincide with the rate of the single species concerned, and the inverse correlation is perfect. Therefore, only the species softness will be used as a weak estimate for novelty: whenever we diagnose lack of erosion due to species softness, we suppress the alternative diagnosis "convergence accumulation", and we state that the pattern is valid even though it may not be "monolithic" due to accumulation. (A pattern that consists of shared novelties and corresponding convergences is not "monolithic", that is, is has two or more sources from which the similar character states originate. In contrast, a pattern due to erosion is "monolithic", but parts of the monolithic whole have eroded away.)

Our estimate is weak anyway, simply because different preservation rates are neither a necessary nor a sufficient condition for erosion (or accumulation) – preservation rates can be highly misleading. In case of erosion, we note the following. Different preservation rates are *not necessary* since random substitutions in individual species may overshadow any difference due to the erosive process itself. For the same reason, they are *not sufficient* either. Moreover, species in one monophyletic group may all have high preservation rates, while the species in the sister group all have low rates, and we may then suspect erosion to explain a pattern that is at least in part due to shared novelties. In other words, any difference in preservation rates may be in parallel (or, "in concordance") with the evolutionary history of the species. (Remember that necessary conditions give us *all candidates* for the instances struck by a specific phenomenon, while sufficient conditions give us the assurance that the phenomenon occurs for a given instance, even though instances may be missed. The phenomenon implies its necessary conditions, and sufficient conditions imply the phenomenon.)

Irrespective of preservation rates, the link between erosion and pattern validity is quite weak in theory. We have already noted that

- Erosion on the one hand, and the formation of shared novelties leading to valid inconsistency patterns on the other hand, may occur in parallel.
- Patterns may be invalid if they are not due to erosion, but due to accumulation.

(See also sections 2.6 and 2.10; we will resume our investigations in section 3.5.2.)

Since calculating all preservation rates is the dominant factor in softness analysis, and their time complexity is $O(\ell q)$, species softness can be obtained in $O(\ell q)$.

3.5 A Stronger Estimate for Pattern Novelty

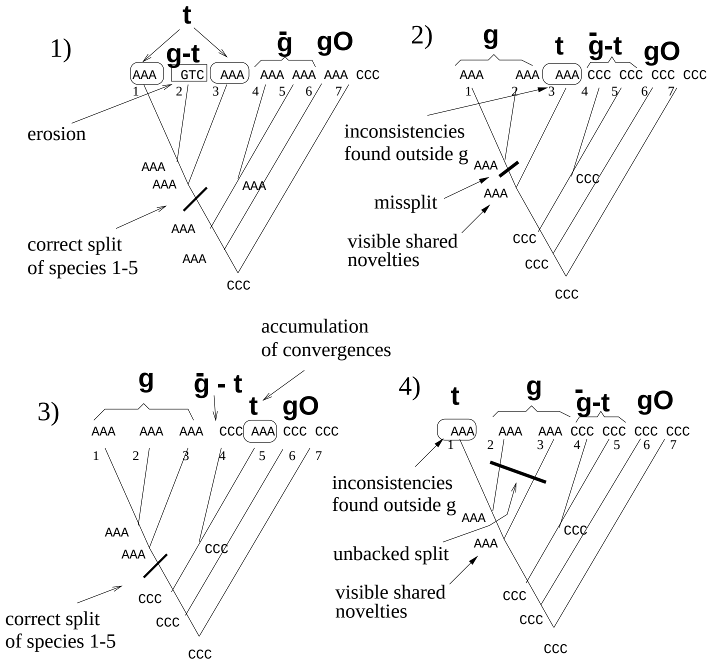
3.5.1 Motivation

In this section we are given a split, an outgroup, and an inconsistency pattern for this split, and we investigate whether the pattern can be explained by erosion, or not. If a pattern cannot be explained by erosion, the novelty estimate we introduce will be maximum.

Consider again Fig. 3.12, panel 1, repeated in Fig. 3.14, panel 1, and the correct split of group 1-5, $G = g$ v $\overline{g}$ = 1-3 v 4,5, with outgroup $gO = 6$. In group 1-3, we find the pattern t = "1,3". At first sight, many occurances of the "1,3" pattern type contribute evidence to the hypothesis that there are character states shared due to exclusive common heritage in a group of species that is composed of 1,3 and the majority of species in 4,5. In other words, there seem to be shared novelties that are torn apart by the split 1-3 v 4,5. For example, this would be the case if there were a monophyletic group 1,3-5. *However*, there are many matches between the inconsistencies in 1,3 and the outgroup 6. We conclude that shared character states in 1,3-5 are *old*. Due to erosion in species 2, the short branches 1,3-5 attract each other. In panel 2, the situation is completely different: pattern "3" in 1,2 v 3-5 does not match the outgroup. The same holds for pattern "1" in 2,3 v 1,4,5 in panel 4 (lower right). Both patterns are valid, due to the *shared novelties* in 1-3. Pattern 5 in 1-3 v 4,5 in panel 3 (lower left) does not match either, even though it is invalid; if there is no erosion, there may still be convergence accumulation. We already confessed in section 2.10 that detecting convergence accumulation is "future work".

There are more immediate challenges, however, which are explained in Fig. 3.15, panels 1 to 4. In the first panel (cf. the first panel in Fig. 3.14), the outgroup 6 was subject to erosion, in the form of modifications eliminating the

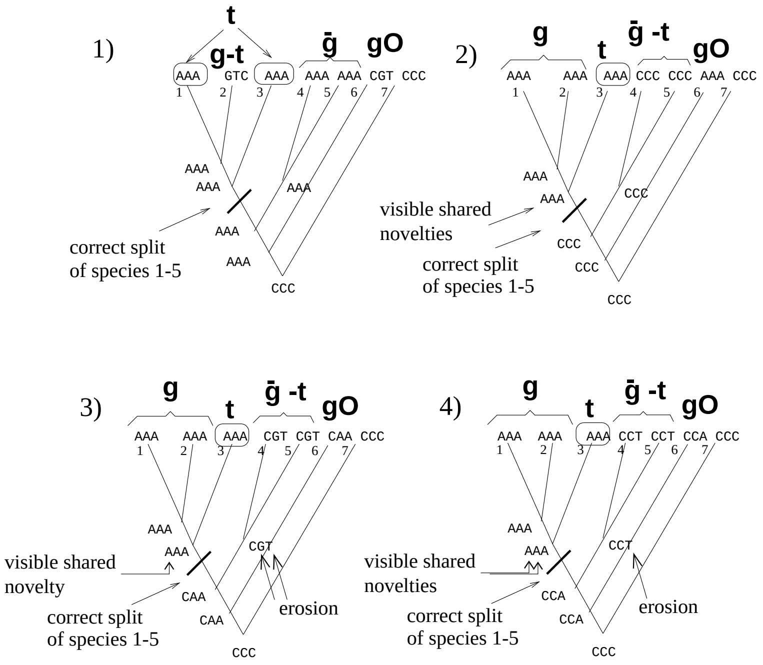
Figure 3.14: Interpreting inconsistency patterns, II.



matches between the pattern t and the outgroup that we observed before. In the second panel (cf. the second panel in Fig. 3.14), the outgroup 6 was subject to convergences. In both cases, the matching rate does no longer carry the information needed to establish erosion correctly: In panel 1, we do no longer suspect erosion in species 2, whereas in panel 2 we are led to suspect erosion in species 4,5. There is, however, an easy trick that alleviates these problems in a lot of cases: Instead of relying on a single matching rate, we calculate two matching rates.

The first matching rate is based on the columns displaying the pattern, and the second is based on columns without the pattern. Consider Fig. 3.16. In panel 1, the situation in Fig. 3.14 panels 1 and 2 is reiterated, and we are happy. The "leftover inconsistencies due to erosion in species 2" are all matching the outgroup, and the "inconsistencies due to shared novelties in species 1-3" are not matching. In panel 2, few substitutions do not affect our judgement. In panel 3, the situation in Fig. 3.15 panels 1 and 2 is reiterated, including one further column, and we are very unhappy. Matches with the outgroup have mostly eroded away in the case of "leftover inconsistencies due to erosion in species 2", and convergences in the outgroup have accumulated in the case of "inconsistencies due to shared novelties of species 1-3". In the panels on the right-hand side of Fig. 3.16, matching rates are compared: In panel 4 and 5,

Figure 3.15: Interpreting inconsistency patterns, III.

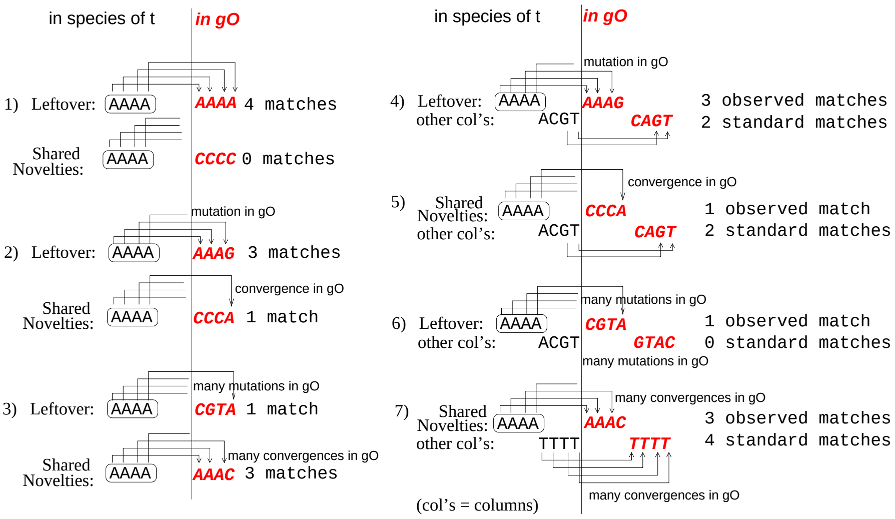


we observe that in general, "leftover inconsistencies due to erosion in species 2" yield a *higher* matching rate than character states in "other" columns that do not display the pattern. On the other hand, "inconsistencies due to shared novelties in species 1-3" yield a *lower* matching rate than character states in "other" columns that do not display the pattern. In panel 6 and 7, we can see that these observations even hold for the problematic cases, ***as long as the "other" columns that do not exhibit the pattern are subject to approximately the same amount of erosion or accumulation in the outgroup.***

Therefore, in Fig. 3.15, panel 1, we compare an "observed" matching rate between the character states of the inconsistency pattern in 1,3 and the outgroup to a "standard" matching rate between character states in 1,3 from other columns not displayed in the tree, and the outgroup, and we expect to observe an *excess* of the observed matching rate, even though there are many modifications in the outgroup. An *excess* of the "standard" matching rate would indicate that there is no erosion. We would expect such an excess in Fig. 3.15 panel 2, as indicated in Fig. 3.16 panel 7. ***This excess will be the novelty estimate, to be defined formally in subsection 3.5.3.*** (The exact definition of "other columns" and the formula for the "standard" matching rate will also be given in subsection 3.5.3).

In section 3.3, we observed that matching rates go up if the outgroup is close,

Figure 3.16: Absolute matching rate versus comparison of matching rates .



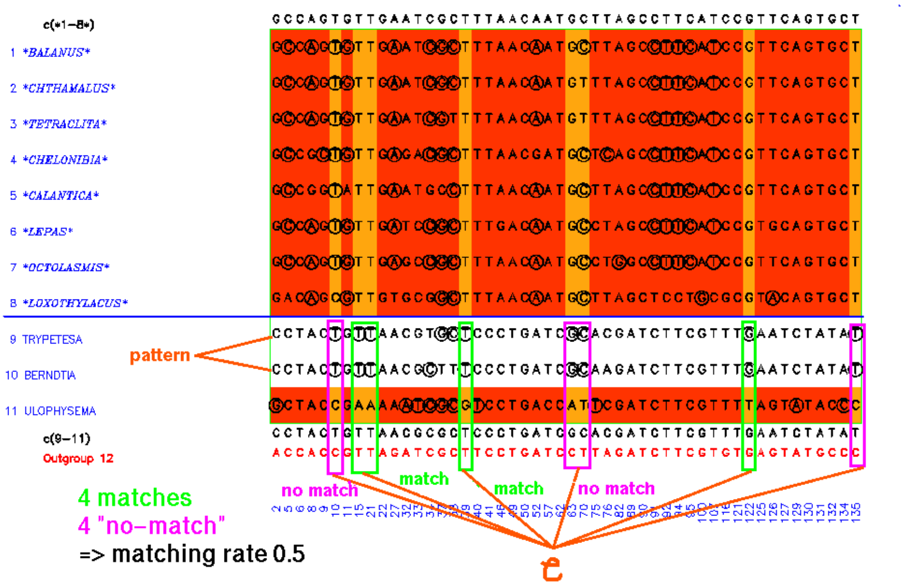
and they go down if it is farther away, but a quotient of matching rates may not be affected much. Exactly this is happening here, if the outgroup was subject to modifications, convergent or not (Fig. 3.15, panel 1 and 2). In panel 1, the process that affected the matching rate of the inconsistency pattern in species 1,3 with the outgroup (that is, erosion of the outgroup 6) has no relation to the process that formed the pattern (that is, erosion of species 2). In panel 2, the process that affected the matching rate of the inconsistency pattern in species 3 with the outgroup (that is, convergence accumulation in the outgroup 6) has no relation to of the process that formed the pattern (that is, the appearance of shared novelties testifying exclusive common heritage in species 1-3). As we will see, processes in panels 3 and 4 of Fig. 3.15 are related, leading to problems to be discussed in the next subsection.

Running Example, Crustacean alignment

Consider Fig. 3.17. For the split 1-8 v 9-11, species 12 is the outgroup supplied by the user. Majority sequences are printed for both groups of species, situated again right above/below the shaded alignment box. As before, circles mark inconsistencies. As we have seen, the character states of the majority sequence of 1-8 form a pattern ”9,10” in columns 10, 15, 21, 39, 63, 70, 122 and 135.

We calculate an ”observed” matching rate of 0.5 for pattern ”9,10” in split 1-8 v 9-11 in Fig. 3.17, and a ”standard” matching rate of 0.818 in Fig. 3.18. An excess of the ”standard” matching rate indicates valid shared novelties:

Figure 3.17: Observed matching rate for pattern ”9,10” in 9-11.



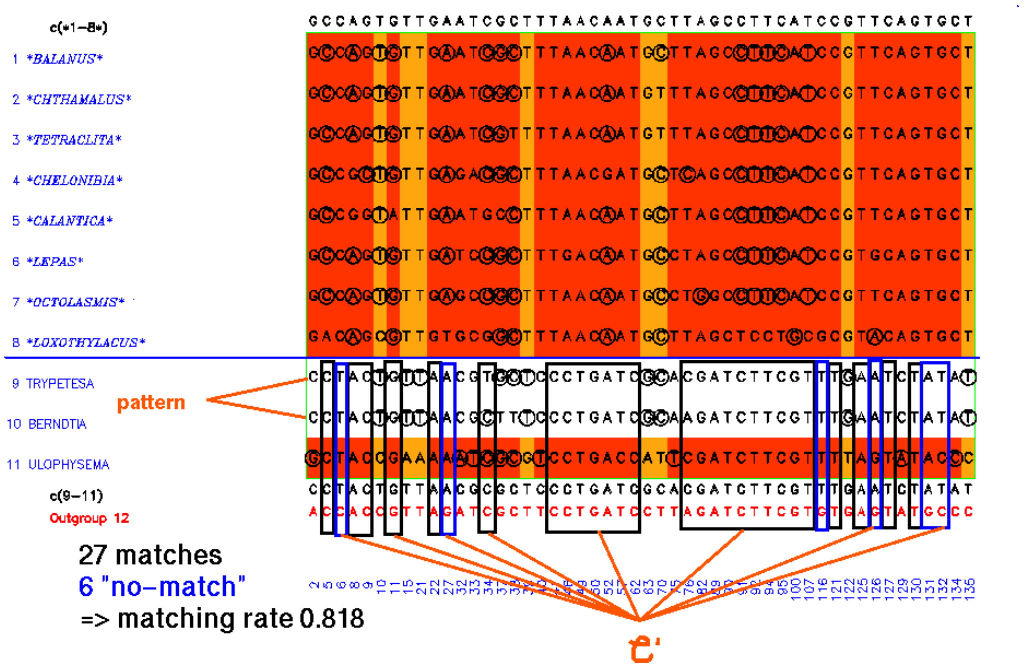
Pattern:	9,10 in 9-11 versus 1-8
Observed Matching Rate:	0.500
Standard Matching Rate:	0.818
Conclusion:	Shared Novelties.

In contrast, observed and standard matching rates for the same pattern in split 1-10 v 11 are displayed in Fig. 3.19 and 3.20. We summarize:

Pattern:	9,10 in 1-10 versus 11
Observed Matching Rate:	0.800
Standard Matching Rate:	0.682
Conclusion:	Erosion.

In the introduction in [8], erosion is discussed in the framework of the two competing splits shown; the observed matching rate of pattern ”9,10” in 1-8 v 9-11 is compared to its observed matching rate in 1-10 v 11. However, there is at least one serious drawback of the direct comparison of observed matching rates. It is not at all guaranteed that the same pattern is actually found in

Figure 3.18: Standard matching rate for pattern "9,10" in 9-11.



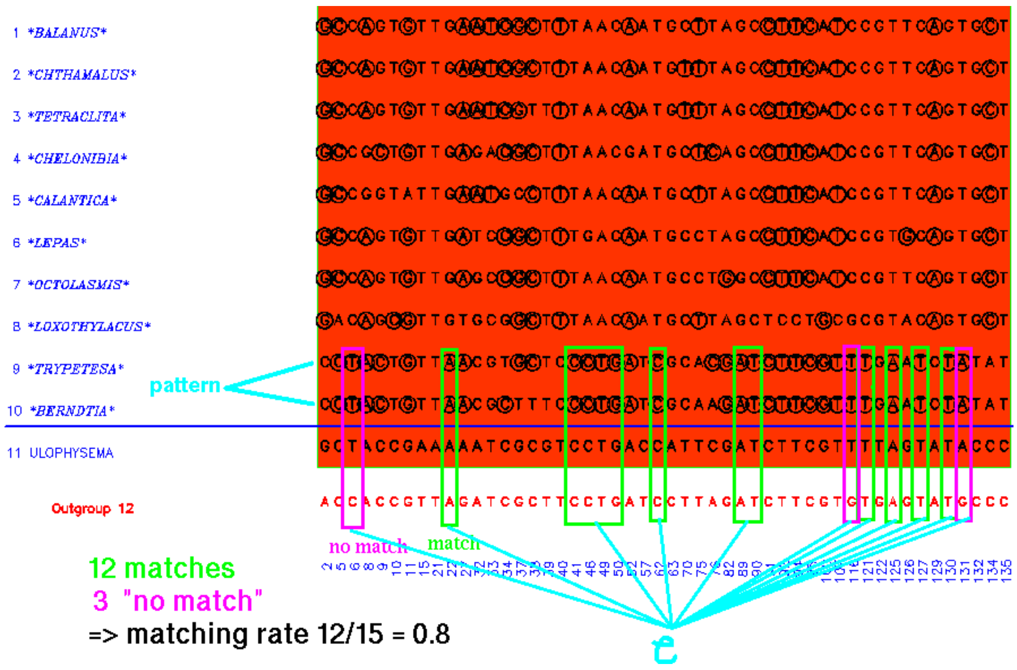
sufficiently many supporting columns in *both* splits. If not, one of the matching rates may be highly unreliable.

3.5.2 Relationships between Validity, Erosion and Matching Rates

In Fig. 3.15, panel 3, one shared novelty in the first alignment column, in species 1-3, triggers a pattern "3" in 1,2 v 3-5. The resulting inconsistency ("A", in species 3) does not match the outgroup ("C"). Furthermore, the outgroup matches with a pattern "3" found in species 1-3 that is due to erosion in species 4 and 5. This process affecting the matching rate of "the inconsistency pattern found in species 1-3" with the outgroup is related to the valid pattern formation process: both trigger pattern "3" ! The inconsistencies are due to both a shared novelty (column 1) and erosion (columns 2 and 3). In panel 4, the situation is similar: the inconsistencies in columns 1 and 2 are due to shared novelties, while the inconsistency in column 3 is due to erosion.

(We already note an observation with respect to the upcoming combination of the weak novelty estimate, that is the species softness, and the novelty es-

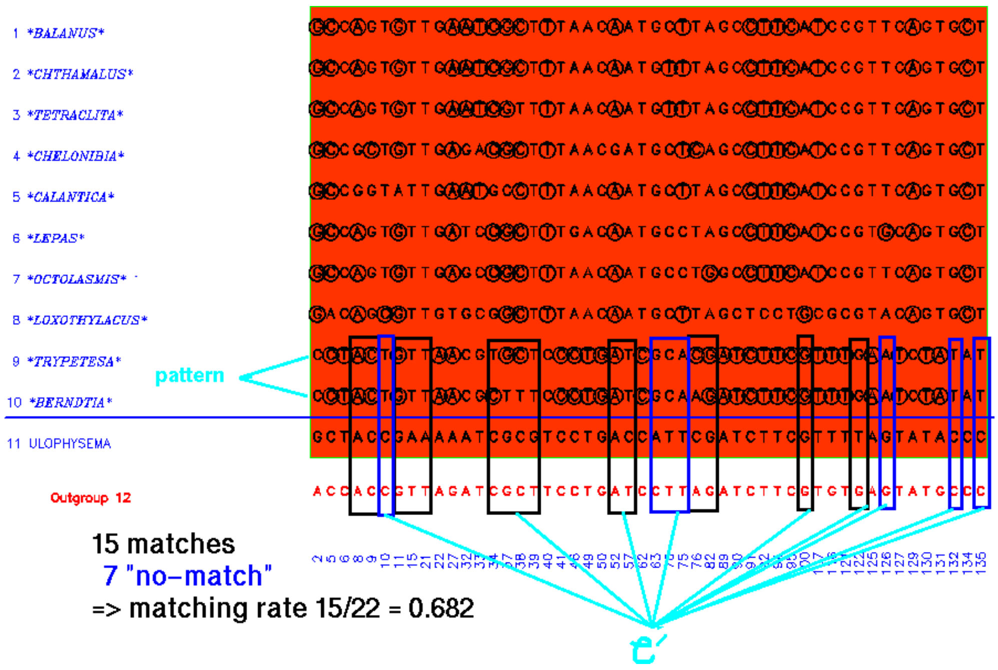
Figure 3.19: Observed matching rate for pattern ”9,10” in 1-10.



timate just described, that is based on outgroup comparison, into one validity estimate. The combination will be designed such that in panel 3, high species softness of the pattern species 3 (derived from columns other than the ones shown), in contrast to species 4,5 would indicate lack of erosion, and it could still render a sufficiently large validity estimate. In panel 4, high softness is the case for the columns shown: it is exactly the shared novelties in 1-3 that reduce the preservation rate of species 3 with respect to the outgroup. As we will see, this reduction can contribute to a correct decision in favor of pattern validity.)

Like the weak novelty estimate, the species softness, the stronger novelty estimate just introduced suffers from inherent problems. In fact, an excess of the observed matching rate is at least not a necessary condition for erosion. It is *not necessary*, because an erosive process may be in parallel with the evolutionary history of the species. As we have seen in Fig. 3.15, panel 4, on page 67, repeated in Fig. 3.21, panel 3, on page 73, the observed matching rate then goes down because there are also inconsistency patterns testifying shared novelties. This is actually *a good thing*, since it is the shared novelties that we want to detect, in spite of erosion. It is not known whether an excess of the observed matching rate is a sufficient criterion – more theoretical work is needed to tackle this issue.

Figure 3.20: Standard matching rate for pattern ”9,10” in 1-10.



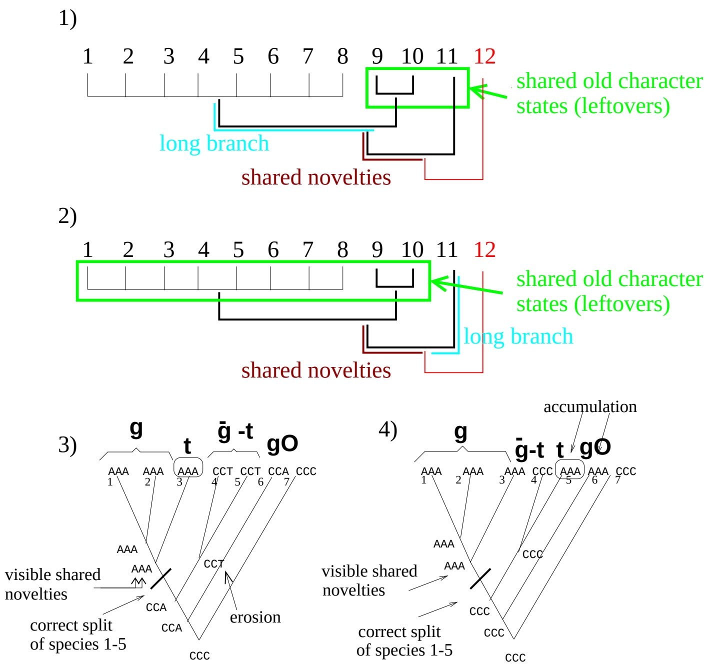
In particular, convergences may accumulate in both the pattern species as well as the outgroup, as displayed in Fig. 3.21, panel 4, on page 73. Such ”twofold” accumulation yields a relatively high observed matching rate that is not due to erosion, but the standard matching rate should be affected as well.

Therefore, the stronger novelty estimate based on outgroup matching rates is much more useful than the weak estimate. It turns out to be susceptible to mixed cases, where both erosion and valid pattern formation by speciation happen, and then it favors the correct answer, whenever it fails as an erosion detector.

Sampling error put aside, we conjecture that the stronger novelty estimate just leaves us with the two main problems noted for our approach:

- Erosion is not a sufficient condition for the *in*validity of patterns. Patterns may have valid and invalid components, if the following happens in parallel:
  - Speciation, testified by the formation of shared novelties leading to valid inconsistency patterns.
  - Branch attraction, that is the formation of shared old character states

Figure 3.21: Shared novelties and leftovers may be observed for the same set of species.



due to erosion.

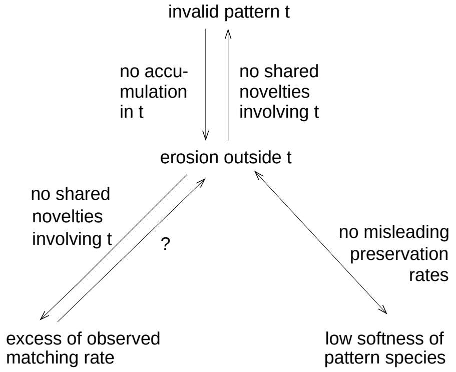
- Erosion is not a necessary condition for the *in*validity of patterns. Patterns may be invalid due to convergence accumulation.

Fig. 3.22 summarizes our conjectures regarding necessary and sufficient conditions. Arrows are assumed to be valid if the phenomenon that labels them holds. For example, an invalid pattern t is deemed the result of erosion if "no accumulation in t" holds true. The label "no shared novelties involving t" is just rephrasing that erosion and speciation do *not* take place in parallel.

3.5.3 Formal presentation

We will now formalize the ideas just discussed. Given a split $G = g \text{ v } \overline{g}$ of species $\{i_1, ..., i_\ell\} \subseteq \{1, ..., m\}$, let us assume that we have obtained the list of patterns $T(g)$ observed in group g in the alignment $A = (a_i)_{i \in \{i_1,...,i_\ell\}} = a_{i_1}, ..., a_i, ..., a_{i_\ell}$ of length q. (The case of group $\overline{g}$ is analogous). We fix a `minimum column count` δ, which is the minimum number of supporting columns that are needed to trigger an investigation of the corresponding pattern. In other words, pattern types are `ignored` if they occur in strictly less than δ alignment

Figure 3.22: Relationships between invalidity, erosion, and matching rates.



columns. Empirically, δ is taken as $\lceil \log_2 vcols \rceil$, where $vcols$ is the number of variable alignment columns. This value is sufficiently low (it is 8.0 for 256 variable columns) that no relevant information should be lost. The minimum column count serves a dual purpose:

- Computation time and memory requirements are reduced significantly.
- Ignored pattern types can serve nicely for the calculation of the standard matching rate, as explained below.

Given an inconsistency pattern type $t = \{i_1, ..., i_k\} \subset g$ found in columns

$$\mathcal{C} = \mathcal{C}(t) \subseteq \{j_1, ..., j_q\},$$

$|\mathcal{C}| \geq \delta$, and an outgroup majority sequence

$$c(gO) = (c_{j_1}(gO), ..., c_{j_q}(gO)),$$

an `outgroup match` occurs in column $j \in \{j_1, ..., j_q\}$ if and only if

$$c_j(a_{i_1}, ..., a_{i_k}) = c_j(gO).$$

For $j \in \mathcal{C}$ we have

$$c_j(a_{i_1, \ldots, a_{i_k}}) = a_{i_1},$$

and we just check

$$a_{i_1} = c_j(gO),$$

since the inconsistencies must all be equal. The `observed matching rate`

$$m = m(t) = m(t, gO, \mathcal{C})$$

is the relative number of outgroup matches in the pattern-supporting columns $\mathcal{C}$, and the `standard matching rate`

$$m_0 = m_0(t) = m(t, gO, \mathcal{C}')$$

is the relative number of outgroup matches in the `standardizing columns` of t,

$$\mathcal{C}' = \mathcal{C}'(t) = \{j \in \{j_1, ..., j_q\} : t' = t(j) = \emptyset \text{ or } t' = t(j) \text{ is ignored }\}.$$

The standard matching rate checks outgroup matches considering the same species t as the observed matching rate, but for a different set of columns. This set $\mathcal{C}'$ consists of columns featuring no pattern type, and columns featuring an ignored pattern type. The former may feature only deviations that are not inconsistent because they do not match with the majority of the other group, or columns that are fixed in g – in both cases, no pattern type is observed. Columns with a different, *unignored* pattern are not used for standardization. They give rise to their very own matching rate. Columns for which there is no majority character state in $\overline{g}$ (such that no inconsistencies can be detected) are not considered either. These are too noisy to contribute any valuable information about the standard matching rate. In particular, these columns must feature gaps in the majority of species in $\overline{g}$, if the minimum invariability threshold τ is 0. If the majority of species in $t \subset g$ features gaps as well, these columns influence the standard matching rate by some arbitrary amount, depending on the number of gaps in the outgroup. Only the standard matching rate is influenced by gaps in the outgroup, but not the observed matching rate. This is because the observed matching rate is established only for inconsistent character states, which must not be gap since they must match with a majority character state.

As the overall number of random deviations in the dataset increases, so does the number of inconsistent ones, and the first subset of $\mathcal{C}'$ shrinks, whereas the second subset grows. If more species are included in a group, we usually observe the same effect, since the new species usually have at least some random deviations.

The criterion for detecting erosion, called the `novelty estimate` n, can now be formalized as the difference between standard matching rate m_0 and the observed matching rate m, activated as explained in section 3.2.3:

$$n = n(t) = excess_{\theta_m}(m_0, m) = \frac{1}{1 + e^{-(m_0 - m)/\theta_m}}.$$

The larger the excess, the more likely *no* erosion took place. The slope-smoothness of the activation is given by the constant θ_m, which is set to the standard 0.1.

In summary, outgroup comparison enables the interpretation of inconsistency patterns. Comparatively high observed matching rates indicate the presence of erosion, comparatively low rates are interpreted as an indication of validity. As noted, the detection of erosion should not rely on the observed matching rate alone, since we would then ignore the "proximity" of the outgroup to the group under investigation. In other words, we use a standardized (or normalized, or calibrated) novelty estimate which is robust to the length of the path to the outgroup.

3.6 Combining Pattern Novelty Estimates into an Activated Validity Estimate

Our next step is to combine the novelty estimate n of a pattern t and its weak, preservation-based species softness q, into one `validity estimate` v, using the *advised* function introduced in section 3.2.2. Empirically, best results are obtained if we limit the impact of the advice even more, by weighting it one half:

$$v = v(t) = \frac{n + advised_{\theta_n}(n, q)}{2}$$

.

The slope-smoothness of the advice activation is given by the constant θ_n, which is again set to 0.1.

Let v_0 be the `acceptable validity estimate` for which a pattern count shall have one half of the full impact. We set $v_0 = 0.75$, as discussed in section 3.2.3. Then, we can activate the validity estimate as follows:

$$\widehat{v} = \widehat{v}(t) = excess^{\bowtie}_{\theta_v,\eta}(v, v_0),$$

using a slope smoothness parameter θ_v, which is again set to 0.1, symmetric scaling, and non-residual activation with exponent $\eta = 5$ (see section 3.2.3).

This `activated validity estimate` $\widehat{v}$ will next be multiplied with the pattern count, which we will activate for reliability.

Finding the list of patterns $T(g)$ for group g taken from a set of species of size $O(\ell)$ takes time $O(\ell q)$ for the majority sequence of $\overline{g}$, and for the subsequent inspection of all subcolumns involving the species in g, where q is the length of the alignment.

The time complexity for calculating the novelty estimate for a pattern t of size $O(\ell)$ is $O(\ell q)$, where q is again the length of the alignment. In detail, the majority sequence takes $O(\ell q)$ for the standardizing columns $\mathcal{C}'$ and $O(\ell q)$ for an outgroup of size $O(\ell)$. It is already given for the supporting columns $\mathcal{C}$. The computation of the rates themselves take an additional $O(q)$. Since the time complexity for softness is the also $O(\ell q)$, we end up with $O(\ell q)$ running time consumed for the validity estimate of one pattern.

In the worst case, the number of patterns is $O(\min(\frac{q}{\log q}, 2^\ell))$ since the minimum column count δ is logarithmic in the number of alignment columns, and for alignment length q, there can be at most $\frac{q}{\delta}$ columns displaying pattern types that occur in at least δ columns. Naturally, the number of patterns cannot be more than exponential in the number of species.

Running Example, Crustacean alignment

Returning to Fig. 3.17, the pattern type featured most in group 9-11 is "9,10", in columns 10, 15, 21, 39, 63, 70, 122 and 135. Next in line is pattern type "11", found in columns 2, 32, 33, 40, 75, 129 and 134. Finally, pattern type "9,11" is observed in columns 37 and 38, and "10,11" is observed once in column 34. If we set the minimum column count threshold δ to 2, the results of the running example are well in line with the results of the Crustacea data from which it is derived by retaining the upfront columns (see section 4.4.1). The pattern "10,11" is then ignored since it is observed only once. For the first group 1-8, a total of 10 different pattern types can be observed. The most prominent one is "1-7" in columns 5, 10, 27, 91, 92, 94, 95 and 107.

We can prepare the following table for the first group 1-8 and the second group 9-11. For comparison, we also append parts of the table for group 1-10 in split 1-10 v 11 (cf. Fig. 3.19). Numbers marked by an asterix ($^{(*)}$) are adjusted by considering the impact of "neighboring" patterns as discussed in section 3.12.

Patt. type t	Found in Columns	Pattern Count s	Observed Matching Rate m	Standard Matching Rate m_0	Novelty n	Soft-ness q	Validity Estimate v	Activ'd Validity $\widehat{v}$	Activ'd Count $\widehat{s}$	Pattern Conflict $\widehat{s_v}$
Group 1-8										
1-7	5 10 ...	8 (8.182$^{(*)}$)	$\frac{6}{8} = 0.756^{(*)}$	$\frac{15}{36} = 0.417$	0.033	0.198	0.021	0.000	8.182$^{(*)}$	0.000
... 9 more patterns, 6 ignored, 3 with an activated validity estimate $\widehat{v} = 0$ ...										
Group 9-11										
9,10	10 15 ...	8	$\frac{4}{8} = 0.500$	$\frac{27}{33} = 0.818$	0.960	0.802	0.975	1.000	4.000	4.000
9,11	37 38	2	$\frac{2}{2} = 1.000$	$\frac{26}{33} = 0.788$	0.107	0.198	0.064	0.000	0.000	0.000
10,11	34	1	–	–	–	–	–	–	–	–
11	2 32 ...	7 (7.616$^{(*)}$)	$\frac{6}{7} = 0.869^{(*)}$	$\frac{27}{33} = 0.818$	0.376	0.198	0.253	0.000	0.000	0.000
Group 1-10										
9,10	6 22 ...	15 (15.888$^{(*)}$)	$\frac{12}{15} = 0.811^{(*)}$	$\frac{15}{22} = 0.682$	0.215	0.057	0.127	0.000	15.888$^{(*)}$	0.000
... 17 more patterns, 14 ignored, 3 with an activated validity estimate $\widehat{v} = 0$ ...										

This table follows up on all the steps of the MCOPE cascade as introduced in section 3.1. Discussion of the last two table columns, the activated count $\widehat{s}$, and the pattern conflict (the amount of evidence for shared novelties, $\widehat{s_v}$), will be postponed until section 3.8.

Let us exemplify how we arrive at the validity estimate of pattern "9,10" in group 9-11, given species 12 as the outgroup (Fig. 3.17). We observe this pattern in 8 columns. In columns 15, 21, 39 and 122, the inconsistent nucleotide matches the outgroup. In columns 10, 63, 70 and 135 it does not. Therefore, the observed matching rate is set to $0.500 = \frac{4}{8}$. Our standard matching rate is based on 33 columns (see Fig. 3.18); these are either fixed in 9-11 (like columns 5, 6, 8, 9), display only consistent nonconvergent deviations (columns 57, 76, 126, 132), or they display a pattern type that is ignored because it occurs in an insufficient number of columns (column 34). Of these 33 columns, 27 match the outgroup, and the standard matching rate is $0.818 = \frac{27}{33}$. Activating this difference, we obtain a novelty estimate of

$$n = excess_{0.1}(0.818, 0.500) = 0.960.$$

The softness of the species involved in our pattern, species 9 and 10, is

$$q = excess_{0.1}(0.760, 0.620) = 0.802,$$

because the maximum preservation rate of the species involved in the pattern, that is species 9 and 10, is 0.620 and the (minimum) preservation rate in the complement, species 11, is 0.760 (cf. Fig. 3.11 on page 61). Taking the advice, we obtain a validity estimate of

$$v = 0.5 \cdot 0.960 + 0.5 \cdot \mathit{advised}_{0.1}(0.960, 0.802) = 0.5 \cdot 0.960 + 0.5 \cdot 0.989 = 0.975.$$

Finally, the activated validity estimate is

$$\widehat{v} = \mathit{excess}^{\bowtie}_{0.1,5}(0.975, 0.750) = 1.$$

The validity estimate for pattern ”9,10” in 1-10 v 11 is completely different (see Fig. 3.19). This time, the observed matching rate is based on 12 matches in columns 22, 41, 46, etc, and 3 mismatches in columns 6, 116 and 131. This yields an observed matching rate of 0.800, which is adjusted to 0.811, considering ”neighboring patterns” as explained in section 3.12. This time, the standard matching rate is based on 22 columns, 15 of which match, yielding a rate of 0.682 (see Fig. 3.20). Activating this difference, we obtain a novelty estimate of

$$n = \mathit{excess}_{0.1}(0.682, 0.811) = 0.215,$$

which will take the conforming advice of

$$q = \mathit{excess}_{0.1}(0.340, 0.620) = 0.057,$$

yielding a validity estimate of

$$v = 0.5 \cdot 0.215 + 0.5 \cdot \mathit{advised}_{0.1}(0.215, 0.057) = 0.127,$$

which is then squashed by the sigmoid activation:

$$\widehat{v} = \mathit{excess}^{\bowtie}_{0.1,5}(0.127, 0.750) = 0.$$

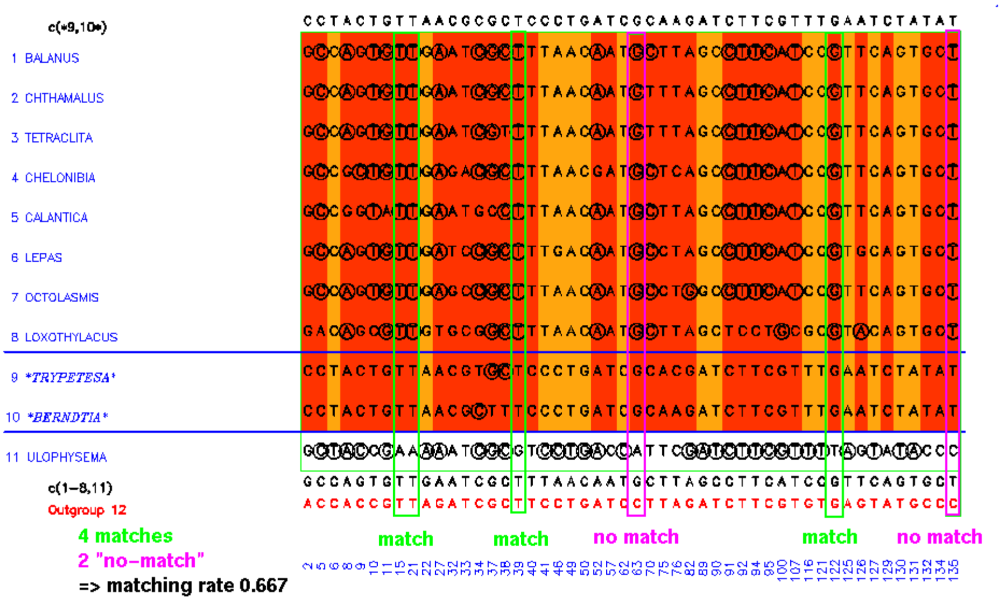
(Note that non-conformant advice would not be taken; e.g. $\mathit{advised}_{0.1}(0.215, 1.000)$ evaluates to 0.238, and 0.238 is also squashed to 0.)

The standard matching rate of pattern ”9,11” in 1-8 v 9-11 is $\frac{26}{33}$ instead of $\frac{27}{33}$ since the majority nucleotide C of species 9,11 in column 76 does not match the outgroup, but the majority nucleotide A of species 9,10 in the same column does. Other than that, calculations of the validity estimates for patterns ”9,11” and ”11” in group 9-11, as well as ”1-7” in group 1-8 are straightforward.

We now return to the problems caused by erosion and speciation occuring in parallel, cf. section 3.5.2. Panel 1 of Fig. 3.21 displays the familiar situation of leftovers in species 9-11 and shared novelties in species 1-10, because species 1-8 are ”more evolved”. In contrast, Fig. 3.21, panel 2 indicates that species 11 evolved rapidly as well, and shared old character states (leftovers) as well as shared novelties are then found in species 1-10.

The consequence is that the split 1-8,11 v 9,10 in Fig. 3.23 becomes a borderline case, where the inconsistencies in 1-8 (in columns 15, 21, 39, 63, 122 and 135) are due to shared novelties as well as erosion, and they match the outgroup even more often than the character states in other columns. However,

Figure 3.23: Observed matching rate for pattern ”1-8” in 1-8,11.



the softness of the pattern-forming species 1-8, in contrast to species 11, gives the much-needed advice that a lack of erosion occuring in species 11 can be established for the formation of the pattern ”1-8”, and the inconsistencies are therefore considered valid. For our sample alignment, the softness of the pattern species 1-8 is

$$q = q(\text{”}1\text{–}8\text{”}) = excess_{0.1}(\min_{i\in\text{”}11\text{”}} p(i), \max_{i\in\text{”}1-8\text{”}} p(i)) = excess_{0.1}(0.760, 0.480) = 0.943,$$

but the incorrect verdict of a novelty estimate of 0.194, based on an observed matching rate of 0.667 and a standard matching rate of 0.500, is too unambigous to be advised by it. Analyzing the full alignment, however, reveals a novelty estimate of 0.474, which can be advised.

In other words, it is noted as evidence for validity that despite the softness of 1-8, there are still many inconsistencies that do not match, and such a borderline case found for the full-size alignment is then decided in favor of validity.

3.7 Activating a Pattern Count Based on its Reliability

Given a split $G = g \text{ v } \overline{g}$, and an inconsistency pattern type $t = \{i_1, ..., i_k\} \subset g$, found in columns $\mathcal{C} = \mathcal{C}(t)$, recall that $|\mathcal{C}| = |\mathcal{C}(t)|$ is the `pattern count` of the pattern t, which is also denoted by $s = s(t)$.

We call a pattern `unreliable`, if it has too few supporting columns such that the observed matching rate may be distorted easily by very few substitutions, resulting in an incorrect validity estimate. To get rid of unreliable patterns, we will multiply a pattern count s with its excess to an acceptable pattern count s_0. It is a challenge to establish an appropriate `acceptable pattern count` s_0 for which one half of the activated count is given. The following approaches failed:

- Fix s_0 once and for ever as a small constant, say, 32. The problem is that this idea does not scale well with large alignments.

- Fix s_0 as a derivative from the minimum column count δ, such that it is basically logarithmic in the number of variable columns in the alignment. The problem is that the number of variable columns in an alignment depends strongly on the amount of erosion, but s_0 should not be influenced by this.

The approach now presented is the least studied part of MCOPE, but it has some theoretical appeal, and at the same time, it triggered a significant improvement of results.

Let μ and σ be mean and standard deviation of the distribution of *all* pattern counts of inconsistency patterns in group g. This includes the counts of the patterns ignored for outgroup matching rate analysis. If they were distributed according to a Poisson distribution with mean μ, the standard deviation of their distribution would be $\sqrt{\mu}$. This value is used as the `acceptable standard deviation` of the actual distribution. Let the excess of the acceptable standard deviation with respect to the observed standard deviation,

$$\rho = excess_{\theta_\sigma}(\sqrt{\mu}, \sigma),$$

be the `regularity` of the distribution. Regularity is close to zero if σ is very large because there are outliers, but it is close to one if the distribution is well-behaved. The slope-smoothness of the activation, θ_σ, is set to be $4/30$ of $\sqrt{\mu}$. (The same ratio underlies the choice of the activation for validity estimates. This one was set empirically to 0.1 for an acceptable validity estimate of 0.75; $0.1 = 4/30 \cdot 0.75$.)

If there are no outliers, we set $s_0 = \mu + \nu_s \cdot \sigma$, where ν_s is a small number like 5. This is based on the idea that reliability should be assigned to "significant" counts, and "significant" can be expressed in statistical terms as the standard error of the mean, $\mu + \nu_s \cdot \sigma$. However, if there are outliers, σ can become very large, and inflate s_0. Then, we resort to the term $s_0 = \mu + \nu_S \cdot \sigma$, where ν_S is a small number like 1. To accomodate both cases in a smooth way, we set the `acceptable pattern count` to

$$s_0 = \rho(\mu + \nu_s \cdot \sigma) + (1 - \rho)(\mu + \nu_S \cdot \sigma).$$

Given the observed pattern count s, we can calculate the `activated pattern count` in the following way:

$$\widehat{s} = \widehat{s}(t) = excess_{\theta_s,\eta}(s, s_0) \cdot s,$$

using slope smoothness parameter θ_s and non-residual activation with exponent $\eta = 5$, as described in section 3.2.3 starting on page 51. The slope smoothness parameter θ_s is 4/30 of the acceptable pattern count s_0, analogous to the ratio used for validity estimates, where the slope smoothness of 0.1 is 4/30 of the acceptable validity estimate of 0.75.

3.8 The Pattern Conflict

If a pattern has an activated validity estimate of $\widehat{v}$, and an activated count of $\widehat{s}$, the `pattern conflict`, also called the `amount of evidence for shared novelties`, denoted $\widehat{s_v}$, is then given by

$$\widehat{s_v} = \widehat{s_v}(t) = \widehat{s} \cdot \widehat{v} = excess_{\theta_s,\eta}(s, s_0) \cdot s \cdot excess^{\bowtie}_{\theta_v,\eta}(v, v_0) = evidence^{\bowtie}_{\eta}(s, v).$$

In summary, we modify the pattern count according to its reliability and its corresponding validity estimate by employing a sigmoid activation function as shown in Fig. 3.9 on page 56. Its main features are that neither low validity estimates nor low pattern counts give rise to a valid pattern – the latter guards against small sampling errors that distort the validity estimates of unreliable patterns.

The most time-consuming part of the pattern count activation process is the calculation of the regularity of the distribution of pattern counts. It is linear in the number of all patterns, which is at most $O(\min(q, 2^\ell))$. The calculation needs to be done once per subgroup.

Running Example, Crustacean alignment

Let us return to the sample alignment. The distribution of pattern counts in group 9-11 of split 1-8 v 9-11 is given in column 3 of the table on page 77, listing all four patterns found: "9,10", "9,11", "10,11" and "11". The average pattern count is $\mu = 4.654$ (4.5 without neighbors), and the standard deviation is

$$\sqrt{\frac{(8-4.5)^2 + (2-4.5)^2 + (1-4.5)^2 + (7-4.5)^2}{4}} = 3.041,$$

that is

$$\sqrt{\frac{(8-4.654)^2+(2-4.654)^2+(1-4.654)^2+(7.616-4.654)^2}{4}} = 3.177$$

with neighbors. Since the expected standard deviation is

$$\sqrt{\mu} = \sqrt{4.654} = 2.157,$$

we obtain a regularity of

$$\rho = excess_{0.288}(2.157, 3.177) = 0.028,$$

where $0.288 = (4/30) \cdot 2.157$. Therefore, the acceptable pattern count is

$$\begin{aligned} s_0 &= \rho(\mu + \nu_s \cdot \sigma) + (1-\rho)(\mu + \nu_S \cdot \sigma) \\ &= 0.028 \cdot (4.654 + 5 \cdot 3.177) + (1 - 0.028) \cdot (4.654 + 1 \cdot 3.177) \\ &= 0.028 \cdot 20.537 + (1 - 0.028) \cdot 7.831 \qquad = 8.188. \end{aligned}$$

Now we can evaluate the reliability of the pattern count of pattern ”9,10” in group 9-11. We observe this pattern $s(”9, 10”) = 8$ times. Hence, the activated pattern count evaluates to

$$\widehat{s}(”9, 10”) = excess_{1.092,5}(8, 8.188) \cdot 8 = 4.000,$$

where $1.092 = (4/30) \cdot 8.188$.

Finally, the pattern conflict is

$$\widehat{s_v}(”9, 10”) = evidence^{\bowtie}_5(8, 0.975) = \widehat{s} \cdot \widehat{v} = 4.000 \cdot 1.000 = 4.000.$$

3.9 The Split Conflict

In this section, we are given a split $G = g \text{ v } \overline{g}$, and an outgroup. The lower the maximum pattern conflict in a split is, the more confident we can be that the split is the correct division of the species into two monophyletic groups, because no shared novelties within $g \cup \overline{g}$ are torn apart, resulting in a valid inconsistency pattern.

We usually need to investigate two tables of pattern types, one per group. From both tables, the `split conflict` is the maximum pattern conflict taken over all patterns:

$$conflict(g \text{ v } \overline{g}) = \max_{t \in T(g) \cup T(\overline{g})} \widehat{s_v}(t).$$

Finding the split with minimum conflict is the open issue that we will tackle in section 3.13. Finally, the `species conflict` of a species $i \in g$ can be defined based on the set of patterns into which it is involved,

$$T(i) = \{t \in T(g) : t \cap i \neq \emptyset, \text{ and } t \text{ is not ignored } \}.$$

Species conflict is the average pattern conflict of the patterns that the species is involved in:

$$conflict(i) = \left(\sum_{t \in T(i)} \widehat{s_v}(t) \right) /|T(i)|.$$

The species conflict is used as the row score for the color coding in the figures displaying the inconsistency analysis in an alignment of the "Running Example", and for the *heuristic* search for minimum conflict.

Looking back at all the calculations done to estimate the conflict, we note that during one step a specific influence can be made by the species sampling in the dataset, like the inclusion of closely related or even identical sequences. This step is the calculation of the relative majority sequences, which depends on the composition of the set of species for which the majority is estimated. In the other steps, the inclusion of identical sequences does not pose a threat; for example, patterns just include one species more, if an identical sequence is added to the pattern-forming set of species, and conflict calculations run unaltered for the larger pattern.

Given a split $G = g \text{ v } \overline{g}$, the overall running time of the conflict calculation can be estimated as follows. First, it suffices to consider the calculations done for one subgroup, say g, of size $O(\ell)$. We first find all patterns in time $O(\ell q)$ (see section 3.6). Regularity, and the acceptable pattern count take $O(\min(q, 2^\ell))$ (section 3.8). The maximum number of unignored patterns is $O(\min(\frac{q}{\log q}, 2^\ell))$ (see section 3.6, again). This is the number of validity calculations necessary in the worst case, each of which takes $O(\ell q)$ (as described in section 3.6). Including the final maximization step, which needs to consider all unignored patterns, the overall time consumption then is

$$O(\ell q) + O(\min(q, 2^\ell)) + O(\min(\frac{q}{\log q}, 2^\ell)) \cdot O(\ell q) + O(\min(\frac{q}{\log q}, 2^\ell)),$$

which is dominated by

$$O(\min(\frac{q}{\log q}, 2^\ell)) \cdot O(\ell q),$$

that is

$$O(\frac{q^2}{\log q} \cdot \ell)$$

for sufficiently large ℓ.

Running Example, Crustacean alignment

Inspecting the table on page 77, we observe that $conflict(1-8 \text{ v } 9-11) = 4.000$, whereas $conflict(1-10 \text{ v } 11) = 0.000$.

3.10 Outgroup Maintenance

As discussed in section 3.3, an essential ingredient of our divide-and-conquer method is the maintenance of an outgroup for any set $G = g_1$ v g_2 of species investigated. The character states of this outgroup are the basis of our investigations into preservation and matching rates, resulting in species softness, the novelty estimate and, ultimately, a pattern validity estimate.

MCOPE starts with a group of species G and an outgroup gO as specified by the user. Once G is split into two subgroups $g1$ and $g2$, we need two new outgroups, one for each subgroup. $O(g1)$, the new outgroup for $g1$, may be taken as a nonempty subset of $g2 \cup gO$, and $O(g2)$, the new outgroup for $g2$, may be taken as a nonempty subset of $g1 \cup gO$.

Outgroup calculation is driven by the need

- to stay close to the species in the new group under investigation,
- not to come too close to the species in the new group, and
- not to dilute information by calculating majority sequences among outgroup candidates.

The latter takes care that the outgroup majority stays as informative as possible; majority sequences of $g2 \cup gO$ or $g1 \cup gO$ have been tried without success. The closeness of the outgroup is reflected by the matching rates. It is obvious that a far-away outgroup may not be able to give sufficiently accurate matching rate information. If the outgroup is very distant, matching rates around 0.25 are observed, and the comparison of matching rates tends to be useless. However, if the outgroup is too close, standard matching rates close to 1 are the result, and it becomes impossible for the observed matching rate to exceed the standard one by a sufficient amount. Furthermore, for standard matching rates of, say, 0.9, sampling error can easily diminish the observed matching rate such that no excess is possible even though erosion took place. In other words, outgroup comparison is not very informative if the outgroup is too close, nor if it is too far away.

Therefore, the `homogeneity` of the amount of deviations introduced into a new group g by comparing its species to the outgroup candidate g' is a good criterion for outgroup selection. It is defined as

$$p_\Delta(g, g') = 1 - \left(\max_{i \in g} m(i, g', \{j_1, ..., j_q\}) - \min_{i \in g} m(i, g', \{j_1, ..., j_q\}) \right).$$

For each outgroup candidate g', we calculate the matching rates of the individual species i with the candidate, and then we look at the spread between minimum and maximum. A large spread implies low homogeneity. (Casting our calculation as a comparison of preservation rates, p_Δ is large if there is low spread among the individual preservation rates in g.)

Finally, the outgroup of g_1 is selected by the formula

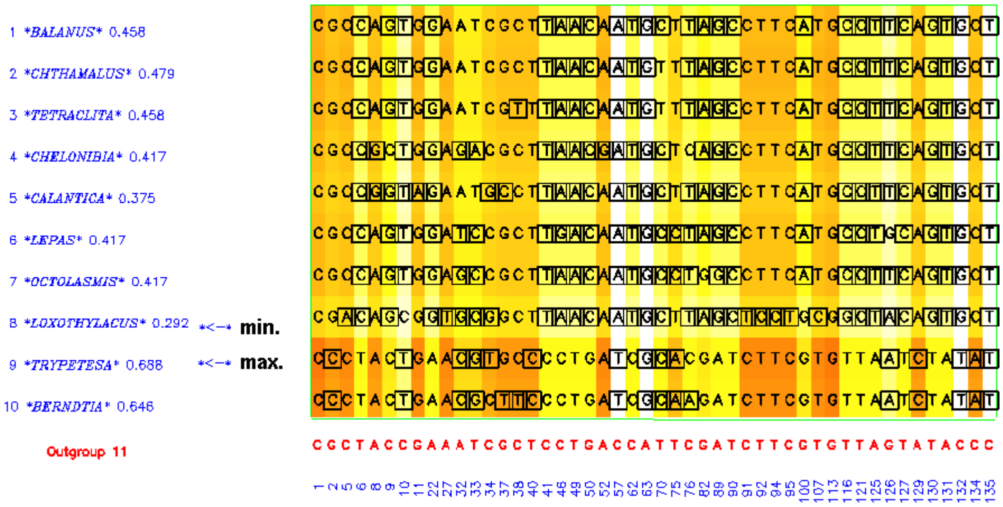
$$O(g_1) = \begin{array}{ll} g_2 & if\ p_\Delta(g_1, g_2) \geq p_\Delta(g_1, gO), \\ gO & otherwise. \end{array}$$

The outgroup of g_2 is selected in an analogous way.

Outgroup maintenance has complexity $O(\ell q)$, since it boils down to the calculation of $O(\ell)$ preservation rates.

Running Example, Crustacean alignment

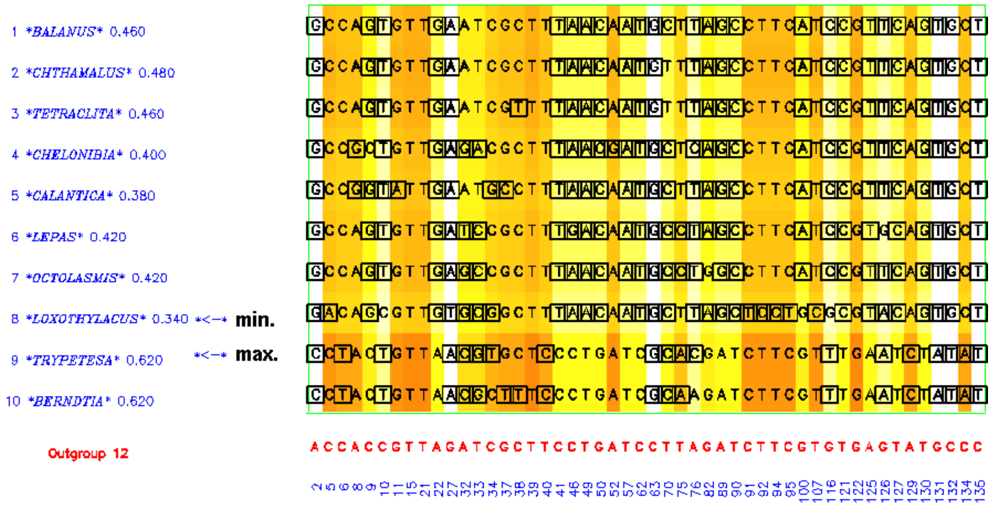
Figure 3.24: Outgroup selection for 1-10; candidate: 11.



We will now explain how an outgroup for group 1-10 can be found, given a choice of "11" and "12". Figures 3.24 (featuring outgroup 11) and 3.25 (featuring outgroup 12) are preservation plots in a similar way as Fig. 3.11; in both cases the first group is 1-10. For outgroup 11, preservation scores have a minimum of 0.292 in species 8, and a maximum of 0.688 in species 9, yielding a homogeneity score of $p_\Delta = 1 - (0.688 - 0.292) = 0.604$. For outgroup 12, preservation scores have a minimum of 0.340 in species 8, and a maximum of 0.620 in species 9 and 10, yielding an homogeneity score of $p_\Delta = 1 - (0.620 - 0.340) = 0.720$. Therefore, outgroup 12 will serve in the next divide-and-conquer step involving species 1-10.

(The acute reader may observe that a different set of columns has been used for candidate 11 versus 12. Columns 1 and 113 are only used for candidate 11, and columns 15, 21 39 and 122 are only used for candidate 12. However, these columns are fixed for 1-10, so they do not influence any differences in preservation rates upon which the selection of the outgroup is based. The curious reader may wish to know why different sets are used. The reason is one of software engineering; the same body of code is used for both preservation rates and outgroup matching rates. Since outgroup matching rates make sense

Figure 3.25: Outgroup selection for 1-10; candidate: 12.



only in the context of a split, a split is ”hidden” in the preservation analysis in the plots of Figs. 3.24 and 3.25 as well. This split is 1-10 v 12 and 1-10 v 11, respectively. In other words, sequence 12 is hidden in the plot for candidate 11, and sequence 11 is hidden in the plot for candidate 12. Both sequences do no harm, but they trigger the display and consideration of columns for which they are variable even if 1-10 are fixed.)

3.11 Detection of Runs

Let us assume that inconsistencies are found in a consecutive sequence of alignment columns. If columns with gaps are not ignored, the suppression of such ”runs” is necessary, because these usually indicate no pattern, but sequencing gaps. For example, in the following initial part of the alignment investigated in section 4.4.6, the pattern type ”1-5,10-12” will occur in columns 1 to 18, if the split 1-12 v 13 is investigated.

```
01 MUS_MUSCU CCTGGTTGATCCTGCCAGTAG-CATATGCTTGTCTCAAAGA
02 ORYCTOLAG CCTGGTTGATCCTGCCAGTAG-CATATGCTTGTCTCAAAGA
03 HOMO_SAPI CCTGGTTGATCCTGCCAGTAG-CATATGCTTGTCTCAAAGA
04 XENOPUS_L CCTGGTTGATCCTGCCAGTAG-CATATGCTTGTCTCAAAGA
05 LATIMERIA CCTGGTTGATCCTGCCAGTAG-CATATGCTTGTCTCAAAGA
06 SQUALUS_A -------------------AG-CATATGCTTGTCTCAAAGA
07 ECHINORHI -------------------AG-TATATGCTTGTCTCAAAGA
08 FUNDULUS_ -------------------AG-CATATGCTTGTCTCAAAGA
09 SALMO_TRU ------------------TAG-CATATGCTTGTCTCAAAGA
10 PETROMYZO CCTGGTTGATCCTGCCAGTAG-CATATGCTTGTCTCAAAGA
11 LAMPETRA_ CCTGGTTGATCCTGCCAGTAG-CATATGCTTGTCTCAAAGA
12 BRANCHIOS CCTGGTTGATCCTGCCAGTAGTCATATGCTTGTCTCAAAGA
13 SACCOGLOS CCTGGTTGATCCTGCCAGTAGTCATATGCTTGTCTCAAAGA
```

Since we would like to "flag" (identify) the indices involved in runs in a flexible way, the following routine takes the following set of parameters:

- ρ_δ, the minimum index difference that defines "consecutive",
- ρ_β, the minimum number of consecutive indices of a "long run" which is flagged from start to end, and
- ρ_α, the number of consecutive indices that are *not* flagged at the start of a "short run" of length smaller than ρ_β.

`Strip-Runs`, flexible identification of runs.
Input: A sequence of indices $J \subset \{j_1, ..., j_q\}$, representing the columns that support an inconsistency pattern, and a set of parameters.

```
• $artifact\_cols = [\,]$ /* empty list */
• $tmp = -\rho_\delta - 1$
• $tmplist = [\,]$ /* empty list */
• foreach index $j \in (J, \infty)$ do
    • if $j - tmp < \rho_\delta$ do /* run starts or continues */
      $tmplist = \left( tmplist, \begin{array}{ll} (tmp, j) & \texttt{if } tmplist = [\,] \\ j & \texttt{else} \end{array} \right)$
      else /* run stops */
      if $tmplist \neq [\,]$ do
        • if $|tmplist| \geq \rho_\beta$ $artifact\_cols = (artifact\_cols, tmplist)$
          else $artifact\_cols = (artifact\_cols, tmplist_{\rho_\alpha..|tmplist|})$
        • $tmplist = [\,]$
    • $tmp = j$
```

`Strip-Runs` returns $artifact\_cols$ as the list of indices involved in runs.

In the foreach loop, we first compare the first index and $-\rho_\delta - 1$; this comparison must fail (i.e. $j - tmp < \rho_\delta$ does not hold for $tmp = -\rho_\delta - 1$.) Therefore, the first index is copied into *tmp*. Then, the copied index is compared to the next one, which is then copied as well, etc. As soon as a comparison succeeds, i.e. it results in a difference that is smaller than ρ_δ, both current indices (*tmp* and j) are added to the temporary list of flagged indices, *tmplist*. Thereafter, only the new index j needs to be added, as long as the comparison succeeds. Once the comparison fails, the temporary list of flagged indices is concatenated to the global list, and is reset. The symbol ∞ is used as a stop sign, which triggers the last concatenation if there is a run in the end.
`Strip-Runs` is usually called with the following parameters: $\rho_\delta = 2, \rho_\beta = \infty$, and $\rho_\alpha = 4$. Setting $\rho_\beta = \infty$ implies that for every run, the first ρ_α indices are not flagged.

3.12 Involving "Similar" Patterns

3.12.1 Neighbor Patterns and Transformations

Given a set of pattern types, we now introduce a similarity score proportional to the likelihood that one pattern type is just the transformation of the other. A `transformation` consists of one or more `transformation steps`. These are either the loss of an inconsistency in a species, or the gain of an inconsistency via a convergent substitution. If the hamming distance is small, very few random mutations may have transformed one pattern of inconsistencies into the other. If the transformation involves species that are "more evolved", even pattern types with a large hamming distance may be related. In particular, a loss of inconsistencies due to erosion in "more evolved" sequences may transform one pattern into another. In summary, the similarity of two pattern types depends on their hamming distance and on the erosion score, that is an estimate for the likelihood that they are related due to erosion or accumulation, independently of their hamming distance.

Formally, a transformation between two pattern types involves the species that make up the symmetric difference of the two pattern types, as follows. Recall that a pattern type is defined as a list of species taken from a group of species g. As before, a pattern type is written using set notation. Given a split $G = g \vee \overline{g}$ of species $\{i_1, ..., i_\ell\} \subseteq \{1, ..., m\}$, let us investigate group g, without loss of generality. Given pattern types $u, t \in T(g)$, we can define their `symmetric difference` $\Delta(u,t)$ and its complement $\overline{\Delta}(u,t)$ as follows.

$$\Delta(u,t) = \{i \in g : (i \in t \text{ and } i \notin u) \text{ or } (i \in u \text{ and } i \notin t)\}$$

and

$$\overline{\Delta}(u,t) = g - \Delta(u,t).$$

As usual, the norm of the symmetric difference is the `hamming distance`:

$$hamming(u,t) = |\Delta(u,t)|\,.$$

Given a pattern type t, we will investigate its relationship to its `neighbors`, i.e. to the other pattern types u, where $u \in T(g) - \{t\}$.

The following are the "neighbor rules" employed.

1. All neighbors must have smaller pattern counts than the pattern type itself.

2. The pattern and its neighbor do not encompass all species, nor is their intersection empty.

3. "Good" neighbors have a small hamming distance, *and/or* they feature a high softness of the symmetric difference.

4. If the pattern count of a neighbor comes close to the count of the pattern type evaluated, it is downweighted further by up to 50%.

5. Moreover, the weight of the neighbors is multiplied by a user-defined constant, the neighbor impact factor.

6. Finally, if the neighbors display an average observed matching rate very different from the one of the pattern itself, their collective impact is downweighted.

The first four rules restrict the number of neighbors considered. Rules 3 and 4 give rise to a weight, which must pass a certain threshold. This "preliminary neighbor weight" depends on how well rule 3 is met, and on the pattern count of the neighbor (rule 4). The last two "neighbor rules" apply to all neighbors collectively, yielding their overall weight, and count. The overall neighbor count is then added to the pattern count itself.

3.12.2 The Hamming Score

To quantify neighbor rule 3 from the preceding subsection, our goal is a "hamming score" of two pattern types that is the larger, the smaller their hamming distance is. Using sigmoids as usual, it may be expressed as the excess of an "acceptable hamming distance" (to be defined), and the actual hamming distance. In formal terms, let pattern type t and an acceptable hamming distance μ_h be given. For each pattern type $u \in T(g) - \{t\}$, the `hamming score` $\omega_h(u,t)$ is then given as the excess of μ_h and the hamming distance between u and t, as follows:

$$\omega_h(u,t) = excess_{\theta_{\omega_h}}(\mu_h, hamming(u,t)).$$

The slope-smoothness θ_{ω_h} is normally set to 1, a value found empirically. It is midway between the smoothness parameter used for the $[0,1]$-interval, and the smoothness used when activating counts.

If the actual hamming distance is much larger than the acceptable one, the hamming score will tend to zero. If the actual hamming distance is much smaller, the hamming score will tend to one. Therefore, a small acceptable hamming distance has the effect that fewer patterns have a large hamming score. In turn, a large acceptable hamming distance results in more patterns with a large hamming score.

We now turn our attention to the acceptable hamming distance μ_h. Since the size of the group g is the most relevant factor for the evaluation of the hamming distance between patterns in g, the `preliminary acceptable hamming distance` μ_{h0} is set to the $\log_2$ of the size of g. However, the acceptable hamming distance μ_h should also depend on the `sparseness` ρ of the distribution of unignored patterns,

$$\rho = 1 - |T(g, \delta)|/(2^{|g|} - 2),$$

where $|T(g, \delta)|$ is the number of patterns t found in g, with $s(t) \geq \delta$, the maximum of which is $2^{|g|} - 2$. (Recall that δ is the minimum column count.) We note that ρ is close to 1 if few unignored pattern types are observed, and it is close to 0 if most of the possible pattern types are actually realized in more than δ columns. In this case it is likely that neighbors found are not due to the erosion of the pattern type for which neighbors are sought, but instead they are realized like most other patterns, due to the erosion of fixed columns. Hence, the acceptable hamming distance should be decreased. Once again using sigmoids, the formula for the acceptable hamming distance involves activation of the difference between the sparseness and an `acceptable sparseness` ρ_o, and multiplication of the preliminary acceptable hamming distance with the activated sparseness. The `activated sparseness` is

$$\widehat{\rho} = excess_{\theta_\rho}(\rho, \rho_o),$$

where the acceptable sparseness ρ_o is set to 0.1, and the slope-smoothness θ_ρ is the usual 0.1 for values in the $[0, 1]$-interval. Finally, the `acceptable hamming distance` is set to

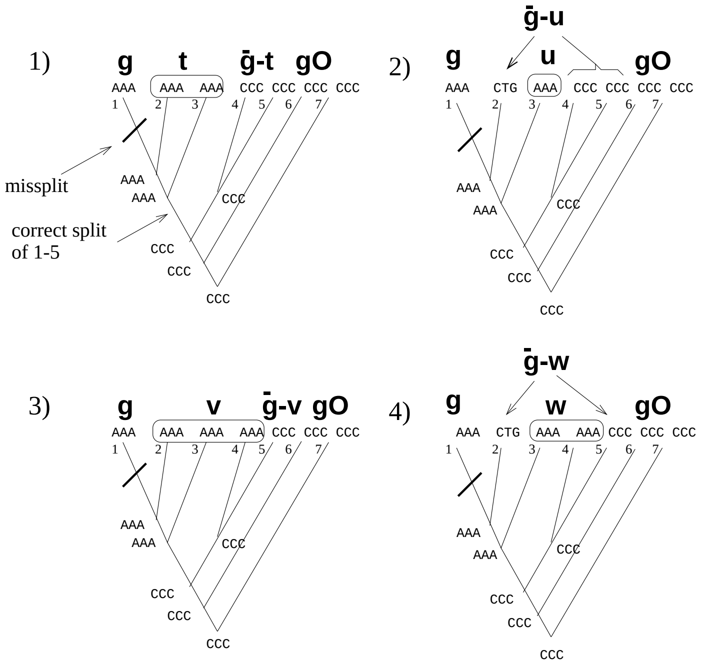
$$\mu_h = \mu_{h0} \cdot \widehat{\rho}.$$

A low acceptable sparseness implies that a low activated sparseness triggers a "significant" reduction of μ_{h0} only if almost all pattern types are found in sufficiently many columns.

3.12.3 The Erosion Score

If the species involved into the symmetric difference of two patterns have low preservation rates, the patterns may be related because one pattern resulted from the other due to an erosive process. In Fig. 3.26, consider pattern $t =$ "2,3" in panel 1 and its neighbor $u =$ "3" in panel 2, where $t \supset u$. If the species in $t - u =$ "2" have low preservation rates, there is evidence that pattern u is derived from pattern t. Conversely, in panel 3, for $v =$ "2-4", $t \subset v$ holds, and

Figure 3.26: Neighbors of a pattern.



we may assume that neighbor v is due to convergences accumulated in $v - t$, if the species in $v - t =$ "4" have low preservation rates. In panel 4, both erosion and acculmulation may have transformed "2,3" into "3,4".

Following up on section 3.4, we define the `erosion / accumulation score`, or `erosion score` for short, of a neighbor as the softness of the symmetric difference

$$q(\Delta(u,t)) = excess_{\theta p}(\min_{i\in\overline{\Delta}(u,t)} p(i), \max_{i\in\Delta(u,t)} p(i)).$$

The slope-smoothness for this activation is θ_p, which is set to the standard 0.1. The erosion score of two patterns is the higher, the more likely one of them is the result of a transformation of the other one, due to erosion.

3.12.4 The Preliminary Neighbor Weight

Up to now, we have focussed our attention on the likelihood of the transformation of one pattern into another. We have also requested that a neighbor cannot have a *larger* count than the original pattern. Another question is how neighbors with a *similar* count should be treated. Giving full weight to these inflates the influence of the neighbors on the original pattern, such that its count and observed matching rate may the completely overshadowed. Our empirical solution to the problem is the introduction of a `size-based downweighting factor`, given by

$$\omega_s(u,t) = excess_{\theta_{\omega_s},\eta}(s(t), s(u)),$$

which uses the same slope smoothness θ_{ω_s} and non-residual activation as the standard pattern count activation. Accordingly, we set the slope smoothness to $(4/30) \cdot s(t)$ and the exponent to 5. Since $s(u) < s(t)$, we have $\omega_s(u,t) > 0.5$. In other words, size-based downweighting can at most amount to 50%.

Given pattern type t, the `preliminary neighbor weight` of neighbor u now evaluates to

$$\omega(u,t) = or_{\theta_{hq}}(\omega_h(u,t), q(\Delta(u,t))) \cdot \omega_s(u,t),$$

using the implementation of a "smooth OR" given in section 3.2.4 with slope smoothness θ_{hq} set to 0.1 as usual.

The following set U of pattern types u with smaller count will be considered for a pattern type t. It meets the neighbor rules 1-4 outlined in subsection 3.12.1.

$$U(t) = \left\{ u \subset g : \left[\begin{array}{l} s(u) < s(t) \text{ and} \\ (t \cap u \neq \emptyset \text{ and } t \cup u \neq g) \text{ and} \\ \omega(u,t) \geq \varepsilon \end{array} \right. \right\},$$

where ε is the `minimum preliminary neighbor weight`. Empirically, ε is set to one half. The first condition maintains that neighbors cannot have more supporting columns than the original pattern ("neighbor rule" 1). The condition $t \cap u \neq \emptyset$ assures a minimum connection between the neighbors in the form of at least one shared inconsistent symbol, and $t \cup u \neq g$ ensures that two pattern types are not just neighbors because they are derived from constant subcolumns which were subject to substitutions at different positions ("neighbor rule" 2).

3.12.5 Influence of Neighbors on the Pattern Count and the Observed Matching Rate

Given neighbors $U(t)$ of a pattern t, we would like to check the difference of

- the observed matching rate of the pattern t, and of
- the combined observed matching rate of the neighbors in $U(t)$.

If this difference is too large, the impact of the neighbors is downweighted, and we will give more credit to the observed matching rate of the original pattern t.

We combine the individual observed matching rates of the neighbors by defining a `weighted average observed matching rate`, given by

$$m(U(t)) = \frac{\sum_{u \in U(t)} m(t, gO, \mathcal{C}(u)) \cdot s(u) \cdot \omega(u,t)}{\sum_{u \in U(t)} s(u) \cdot \omega(u,t)}$$

where $s(u)$ is the number of columns supporting u, and $m(t, gO, \mathcal{C}(u))$ is the matching rate of the majority of the character states in the species making up t with the outgroup, tallied over the columns supporting neighbor u. Taking the majority over the species in t and not in u is in line with the idea that the pattern type u is really a transformation of t.

In contrast to an average observed matching rate, the *weighted* average observed matching rate is calculated by weighting the individual matching rates with the number of columns $s(u)$ on which they are based. Naturally,

$$m(t, gO, \mathcal{C}(u)) \cdot s(u)$$

can be simplified to

$$|\{j \in \mathcal{C}(u) : c_j(t) = c_j(gO)\}|,$$

it is just the number of matching character states.

The `matching rate congruence` of the neighbors of t is now given by

$$\omega_c(t) = excess_{\theta_c,\eta}(0.5, abs(m(t) - m(U(t)))),$$

employing slope smoothness θ_c, and the usual non-residual activation with $\eta = 5$. Empirically, θ_c is set to 0.25, which results in a smoother activation than usual. Such activation is justified because a smoothness of 0.1 would make it impossible to take note of differences $abs(m(t) - m(U(t)))$ less than 0.3:

$$excess_{0.1,5}(0.5, abs(0.8 - 0.5)) = 1.000,$$

whereas

$$excess_{0.25,5}(0.5, abs(0.8 - 0.5)) = 0.581.$$

Novelty scores that differ by $0.8 - 0.5 = 0.3$ should not yield a congruence of 1.000.

Given pattern type t and a `neighbor impact factor` ω_ϵ, the `adjusted pattern count` including neighbors is

$$s^*(t) = s(t) + \omega_\epsilon \cdot \omega_c(t) \cdot \sum_{u \in U(t)} \omega(u, t) \cdot s(u).$$

The adjusted pattern count includes the neighbor counts weighted by the neighbor impact factor, the matching rate congruence, and their individual preliminary neighbor weight.

The `adjusted observed matching rate` of the pattern including neighbors is

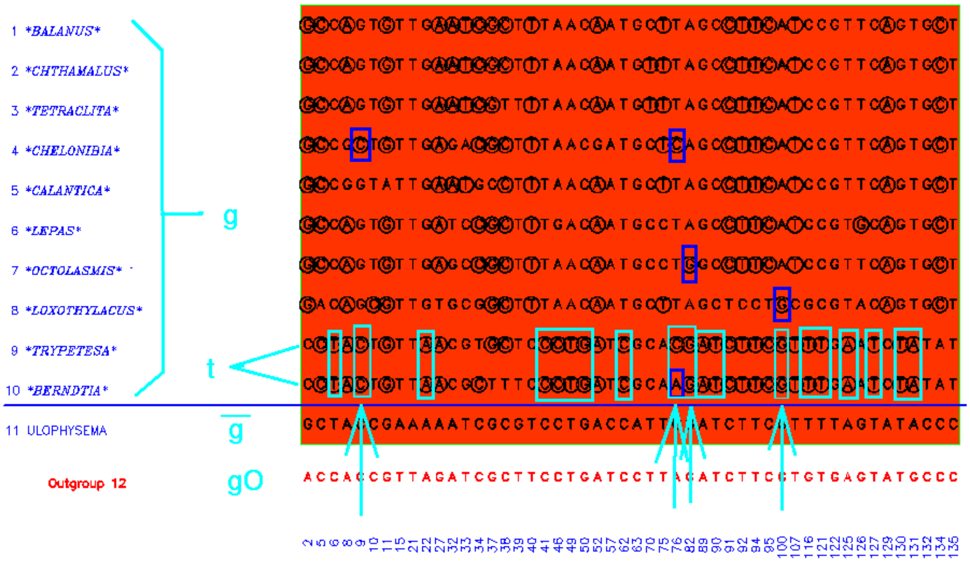
$$m^*(t) = \frac{m(t) \cdot s(t) + m(U(t)) \cdot (s^*(t) - s(t))}{s^*(t)}.$$

Here, we employ exactly the same scheme that we used to compose the adjusted pattern count.

Running Example, Crustacean alignment

Coming back to our running example, some neighbors of pattern "9,10" in group 1-10 taken from split 1-10 v 11 are marked by arrows in Fig. 3.27. They are

Figure 3.27: Neighbors for pattern "9,10" in 1-10.



found in columns 9, 76, 82 and 100. For columns 9, 82 and 100, their hamming distance to "9,10" is 1; pattern "4,9" in column 76 has hamming distance 2. All the other neighbors (like "1,2,4-9" in column 38) do neither feature a small hamming distance, nor does their symmetric difference show signs of erosion.

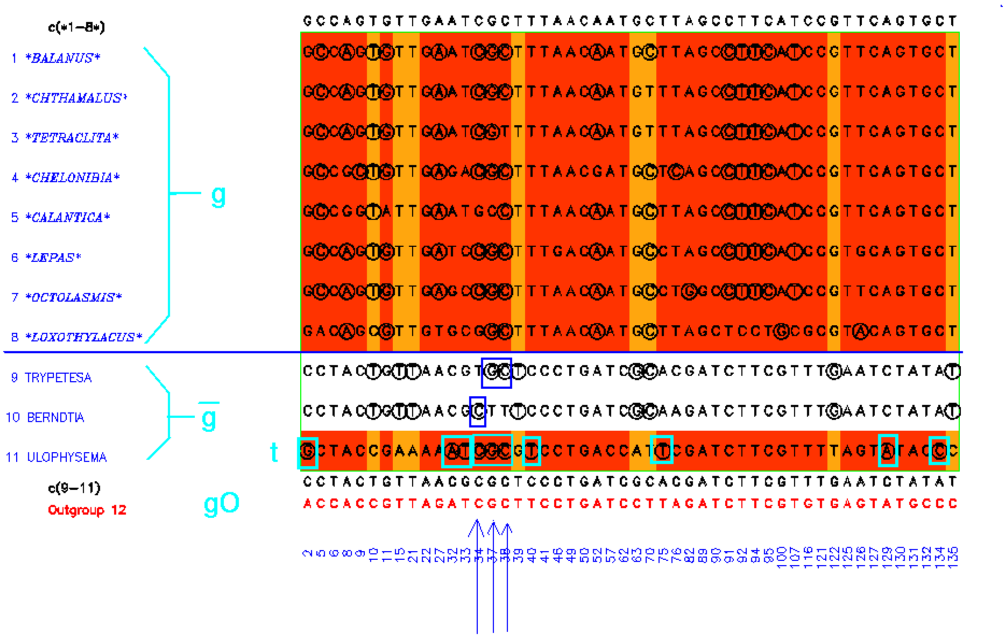
Let us continue by a detailed evaluation of the neighbors of pattern "11" in group 9-11 taken from split 1-8 v 9-11, displayed in Fig. 3.28. We observe this pattern 7 times, in columns 2, 32, 33, 40, 75, 129 and 134 (cf. the table on page 77). Other patterns in group 9-11 are "9,10", "9,11" and "10,11". Pattern "9,10" is not considered as a neighbor since its intersection with "11" is empty. (Another sufficient drop-out reason is its higher pattern count of 8.) Pattern "9,11" in columns 37 and 38 requires more careful evaluation. First, we establish the sparseness ρ of the distribution of investigated patterns,

$$\rho = 1 - |T|/(2^{|g|} - 2) = 1 - 3/(2^3 - 2) = 0.5,$$

which is activated to

$$\widehat{\rho} = excess_{0.1}(\rho, 0.1) = excess_{0.1}(0.5, 0.1) = 0.982.$$

Figure 3.28: Neighbors for pattern ”11” in 9-11.



The preliminary acceptable hamming distance of $log_2 3$ is now multiplied with the sparseness, yielding

$$\mu_h = \mu_{h0} \cdot \widehat{p} = log_2 3 \cdot 0.982 = 1.556.$$

Therefore, the hamming score of ”9,11” with respect to ”11” is the excess of 1.556 compared to the actual hamming distance of 1,

$$\omega_h(u,t) = \omega_h(”9,11”,”11”) = excess_1(1.556, 1) = 0.636.$$

The erosion score of ”9,11” with respect to ”11” is the softness of the symmetric difference ”9” in relation to its complement ”10,11”:

$$q(”9”) = excess_{0.1}(\min_{i \in ”10,11”} p(i), \max_{i \in ”9”} p(i)) = excess_{0.1}(0.620, 0.620) = 0.500.$$

The size-based downweighting factor of the neighbor evaluates to

$$\omega_s(”9,11”,”11”) = excess_{0.933,5}(7,2) = 1.000,$$

where $0.933 = (4/30) \cdot 7$.

The preliminary neighbor weight is larger than the cutoff at $\varepsilon = 0.5$:

$$\begin{aligned} \omega(\text{"}9,11\text{"},\text{"}11\text{"}) &= or_{0.1}(\omega_h(\text{"}9,11\text{"},\text{"}11\text{"}), q(\text{"}9\text{"})) \cdot \omega_s(\text{"}9,11\text{"},\text{"}11\text{"}) \\ &= or_{0.1}(0.636, 0.500) \cdot 1 & = 0.824. \end{aligned}$$

Neighbor "10,11" triggers exactly the same calculations and has the same preliminary neighbor weight.

To find out about the matching rate congruence, we need to evaluate the observed matching rate for the neighbors $U(\text{"}11\text{"}) = \{\text{"}9,11\text{"},\text{"}10,11\text{"}\}$. The average observed matching rate is

$$\begin{aligned} m(U(\text{"}11\text{"})) &= \frac{\sum_{u \in U(\text{"}11\text{"})} m(\text{"}11\text{"}, gO, \mathcal{C}(u)) \cdot s(u) \cdot \omega(u, \text{"}11\text{"})}{\sum_{u \in U(\text{"}11\text{"})} s(u) \cdot \omega(u, \text{"}11\text{"})} \\ &= \frac{1 \cdot 2 \cdot 0.824 + 1 \cdot 1 \cdot 0.824}{2 \cdot 0.824 + 1 \cdot 0.824} & = 1.000, \end{aligned}$$

since the neighbor-supporting columns are all matching.

Matching rate congruence evaluates to

$$\begin{aligned} \omega_c(\text{"}11\text{"}) &= excess_{0.25,5}(0.5, abs(m(\text{"}11\text{"}) - m(U(\text{"}11\text{"})))) \\ &= excess_{0.25,5}(0.5, abs(0.857 - 1)) & = 0.997, \end{aligned}$$

where the matching rate of the original pattern is $m(\text{"}11\text{"}) = \frac{6}{7} = 0.857$. The pattern count including neighbors is

$$\begin{aligned} s^*(\text{"}11\text{"}) &= s(\text{"}11\text{"}) + \omega_\epsilon \cdot \omega_c(\text{"}11\text{"}) \cdot \textstyle\sum_{u \in U(\text{"}11\text{"})} \omega(u, \text{"}11\text{"}) \cdot s(u) \\ &= 7 + 0.25 \cdot 0.997 \cdot (0.824 \cdot 2 + 0.824 \cdot 1) & = 7.616. \end{aligned}$$

Finally, the adjusted observed matching rate of the pattern including neighbors is

$$\begin{aligned} m^*(\text{"}11\text{"}) &= \frac{m(\text{"}11\text{"}) \cdot s(\text{"}11\text{"}) + m(U(\text{"}11\text{"})) \cdot (s^*(\text{"}11\text{"}) - s(\text{"}11\text{"}))}{s^*(\text{"}11\text{"})} \\ &= \frac{0.857 \cdot 7 + 1.000 \cdot (7.616 - 7)}{7.616} & = 0.869. \end{aligned}$$

Both $s^*(\text{"}11\text{"})$ and $m^*(\text{"}11\text{"})$ have already been listed in the table on page 77.

3.13 Exhaustive and Heuristic Searches For Minimum Conflict

Let a set of species $\{1, ..., m\}$ and a split $G = g \text{ v } \overline{g}$ be given. In the preceding sections, we have discussed how we can estimate the split conflict, that is the amount of evidence for shared novelties within $g \cup \overline{g}$ that are torn apart by the split. In this section, our concern is

- the search for a split with minimum conflict, given a set of species, and
- the recursive estimation of a phylogenetic tree.

Informally, we have already outlined our method in figures 1.3 on page 13, and 2.7 on page 36, and we introduced the skeleton of our algorithm at the end of section 2.10 starting on page 42.

As noted, our approach depends on the assumption that the split with the lowest conflict is the correct split, i.e. the one which can be used to identify the largest monophyletic group and its monophyletic complement. We will first present an exhaustive algorithm, and then a heuristic speedup.

The exhaustive version of our algorithm starts with a multiple alignment $S = \{s_{i,j}\}_{i=1...m,j=1...r}$ and calculates the conflict for all $O(2^m)$ splits. Recursively, the two subalignments defined by the split of minimum conflict are subjected again to conflict calculations, and split further, until the sizes of the groups are less than 3. Along the way, a tree is established that has lowest conflict across all its bifurcations.

More precisely, we have the following algorithm.

MCOPE-Exhaustive, phylogeny estimation based on minimum conflict.

Input: A set of species $I = \{i_1, ...i_\ell\}$. We assume that at any time, we have access to an outgroup as described in section 3.10, and to the underlying multiple alignment A. $I = \{i_1, ...i_\ell\}$ are the row indices of the projection of A under investigation.

```
• foreach split $G = g1 \text{ v } g2$ of $I$, $g1 \neq \emptyset \neq g2$, do
    • Calculate $conflict(G)$
• Let $G_{min} = g1_{min} \text{ v } g2_{min}$ be the split such that $conflict(G_{min})$ is minimum over all splits. (Ties are broken arbitrarily.)
• Set $V := \{"g1_{min}"\} \cup \{"g2_{min}"\}$, and
  $E := \{("g1_{min} \cup g2_{min}", "g1_{min}"), ("g1_{min} \cup g2_{min}", "g2_{min}")\}$
• if $|g1_{min}| > 2$, set $(V_1, E_1)$ = MCOPE-Exhaustive($g1_{min}$),
  else if $|g1_{min}| = 2$,
  set $V_1 := \cup_{i \in g1_{min}} \{"i"\}$ and $E_1 := \cup_{i \in g1_{min}} \{("g1_{min}", "i")\}$
• if $|g2_{min}| > 2$, set $(V_2, E_2)$ = MCOPE-Exhaustive($g2_{min}$),
  else if $|g2_{min}| = 2$,
  set $V_2 := \cup_{i \in g2_{min}} \{"i"\}$ and $E_2 := \cup_{i \in g2_{min}} \{("g2_{min}", "i")\}$
• return $V := V \cup V_1 \cup V_2$ and $E := E \cup E_1 \cup E_2$.
```

The inferred tree $\mathcal{T} = (V \cup \{"i_1\text{-}i_\ell"\}, E)$ is returned by MCOPE-Exhaustive. Note that MCOPE-Exhaustive only estimates a topology, but no branch lengths.

MCOPE-Exhaustive is initially called with the set of all species to be investigated, the corresponding alignment, and the user-supplied corresponding outgroup gO.

Since the number of splits is $O(2^m)$, running time of MCOPE-Exhaustive as described will be exponential in m. Therefore, we will *heuristically* try to find a split $G = g1 \text{ v } g2$ of low conflict by listing splits that have high "weight", and of these, we explore the splits that also have low conflict. This will lead to a heuristic algorithm of polynomial complexity. First, however, we will define the notion of the "weak weight" of a split.

An alignment column j provides `weak weight` for a split $G = g1 \text{ v } g2$ if one group exposes a *single* character state that is not displayed by the other group. The `weak weight` of a split $G = g1 \text{ v } g2$ in an alignment $A = \{a_{i,j}\}_{i=1\ldots\ell, j=1\ldots q}$ of length q is the number of weak-weight-providing columns:

$$\begin{array}{lll} & \multicolumn{2}{l}{wgt(g1 \text{ v } g2) = |J|, \text{where}} \\ & \multicolumn{2}{l}{J = \{j \in \{j_1, ..., j_q\} :} \\ \text{there exists } \mathtt{N} \in \mathcal{A}\text{:} & \text{for all } i \in g1 : a_{i,j} = \mathtt{N}, & \text{and for all } i \in g2 : a_{i,j} \neq \mathtt{N}, \\ \text{or} & \text{for all } i \in g2 : a_{i,j} = \mathtt{N}, & \text{and for all } i \in g1 : a_{i,j} \neq \mathtt{N}\}. \end{array}$$

Since $wgt(g1 \text{ v } g2) = wgt(g2 \text{ v } g1)$, it is useful to create a `canonical` list of splits with nonzero weak weight, by keeping only the splits where the last species is not involved in the first group $g1$, thereby removing all duplicates. For an alignment A, the canonical list of splits with nonzero weak weight is called the `weak spectrum`, a weaker notion of the "spectrum" introduced by [9]. (For the case of two character states, there is no difference between our notion and the one by [9]. If there are more character states, we give weak weight to a split even if one group does not feature a single character state.)

Given an alignment $S = \{s_{i,j}\}_{i=1\ldots m, j=1\ldots r}$, of length r, composed of the rows indexed by $\{1, ..., m\}$, we can calculate the set of splits with nonzero weak weight by inspecting each column of S in turn, and for each character state `N` except "$-$" ("gap"), we increment the weak weight of the split that is composed of the species which display `N`, and their complement:

$$\text{for all } j \in \{1, ..., r\}, \text{ for all } \mathtt{N} \in \mathcal{A} :$$
$$\text{if } g = \{i : s_{i,j} = \mathtt{N}\} \neq \emptyset, \text{ increment } wgt(g, \{1, ..., m\} - g) \ .$$

Since the size of the alphabet is constant, this calculation runs in time $O(mr)$. For subalignments A, we can calculate their list of non-zero weak weights by simply projecting the list of weak weights calculated for S: We just add together weak weights for which the indices have the same projection. This list of weak weights can be canonized as described before. Alternatively, we can just recalculate the weak spectrum for each subalignment, and since this renders our running time analysis a bit easier, we will simply assume such recalculations in section 3.14.

We will now introduce the concept of a filter. We introduce two list operators `grep` and `sort`. Given a predicate p, and a list L, $\mathtt{grep}(p, L)$ consists of those elements in L for which p is true. The `sort+` operator just sorts a list in descending order, the `sort-` operator sorts in ascending order. A `max-filter` $f^+(threshold, L)$ is now defined as

$$f^+(threshold, L) := \mathtt{grep}(v > threshold, \mathtt{sort+}(L)).$$

f^+ returns elements larger than *threshold*, in descending order. In contrast, a `min-filter` returns elements smaller than *threshold*, in ascending order:

$$f^-(threshold, L) := \texttt{grep}(v < threshold, \texttt{sort-}(L)).$$

A list of pairs $L = ([u_i, v_i])_{i=1...|L|}$ will always be `grep`ped and `sort`ed according to the second value $v = v_i$; the first value is just carried along.

For each filter in the following discussion, the heuristics also retains a *minimum number of elements*, f_{min}, if necessary by filling up the output list from the best input splits that are not yet included in the list (at least one split is always available, but for $f_{min} > 1$ we may fail to provide the requested number of splits), and the heuristics truncates the output list if a *maximum number of elements*, f_{max}, is exceeded. Implicitly, all filters discussed in the following have f_{min} and f_{max} set to small constants; f_{min} is generally set to 2, and f_{max} is set to 16 (in the case of the starter list based on the weak spectrum), or 8 (in the case of the subsequent list of minimum-conflict splits, see below).

Let $I = \{i_1, ..., i_\ell\}$ be the set of species currently under investigation, and A be the corresponding (sub)alignment. The list of high-weight splits H considered by the heuristics is then based on the weak spectrum, in the form of a list of pairs [split, weak weight]

$$G = ([(g1 \text{ v } g2), \quad wgt(g1 \text{ v } g2)] \quad | \quad wgt(g1 \text{ v } g2) > 0),$$

where $g1 \cup g2 = I$, as follows, given a `minimum weak weight value` μ_{wgt}:

$$H = f^+(\mu_{wgt}, G).$$

μ_{wgt} may e.g. be calculated as 2% of the maximum weak weight value observed.

Then, the list of low-conflict splits H' is based on

$$G' = ([(g1 \text{ v } g2), \quad conflict(g1 \text{ v } g2)] \quad | \quad (g1 \text{ v } g2) \in H)$$

as follows, given a `maximum conflict value` $\mu_{conflict}$:

$$H' = f^-(\mu_{conflict}, G').$$

$\mu_{conflict}$ may e.g. be 20% of the average conflict value of the splits in G'.

Next, the species involved in high-conflict patterns are determined, and, beginning with the species of highest conflict, moved into the other group; conflict improvements are noted, and the optimization process `branches` in this case: it continues the old exploration, with the remaining high-conflict species, and it starts a new one using the new split of lowest conflict and the remaining high-conflict species.

A variant is to move several species in one go, if they were involved in the same valid inconsistency pattern. High-conflict species can be determined using another filter, selecting those candidates that display e.g. more than the average amount of pattern conflict. If we move several species, the list of trailing species has to be updated by removing the piggybacks. For simplicity, we continue discussing the species-by-species case, which is of course less efficient. In all validation runs presented in chapter 4, several species were allowed to move in one go.

Let $G = g1 \text{ v } g2$ be the split currently under consideration. Let $\mathcal{I}$ be derived from the list of species in $g1 \cup g2$ as follows:

$$\mathcal{I} = ([i, \quad conflict(i)] \quad | \quad i \in g1 \cup g2).$$

Then, given a `minimum species exchange conflict value` $\nu_{conflict}$, the list of high-conflict species to be subjected to conflict-based species exchange is

$$\mathcal{I}' = f^{+}(\nu_{conflict}, \mathcal{I}).$$

$\nu_{conflict}$ may e.g. be the average conflict value of the species in $\mathcal{I}$.

The size of H, G', H' and $\mathcal{I}'$ is O(1) since the filters truncate the lists if the maximum number of elements f_{max} is exceeded.

Finally, the following routine is the heuristic to find a split with minimum conflict:

Function `movespecies`, improving a split by conflict-based species exchange.
Input: A split $G = g1 \text{ v } g2$, a list $\mathcal{I}'$ of high-conflict species, and a split $G_{min} = g1_{min} \text{ v } g2_{min}$ with known *conflict*-value.

```
• if $|\mathcal{I}'| = 0$, return $G_{min}$,
  else select the species $i$ with highest conflict from $\mathcal{I}'$
• Set $G := \begin{cases} g1 \cup \{i\} \text{ v } g2 - \{i\} & if\ i \in g2 \\ g1 - \{i\} \text{ v } g2 \cup \{i\} & otherwise \end{cases}$
• if the split $G$ is nontrivial (no group is empty) do:
    • Calculate $conflict(G)$
    • if $conflict(G) < conflict(G_{min})$ do:
        • Set $G_{min} := G$
        • $G_{min}$ = movespecies($G_{min}$,$\mathcal{I}'$,$G_{min}$)
    • $G_{min}$ = movespecies($G$,$\mathcal{I}'$,$G_{min}$)
• return $G_{min}$
```

`movespecies` is called for all splits in H' as the first argument, and a corresponding list of high-conflict species, providing the best split found so far as the third argument; initially this is the first element of H'. The *conflict*-values are stored in a table since splits may be encountered more than once during different calls to `movespecies`, but with different trailing lists of species still to be checked for moving. As a final step, if G_{min} does not appear in H', it should also be checked for any further improvement.

We obtain a heuristic variant of MCOPE-`Exhaustive`, called MCOPE-`Heuristic`, by starting with a list of high-weight splits, finding the low-conflict splits of these, and then using `movespecies` as a search routine to find a split of lowest conflict.

It is not clear whether the current main bound on the number of heuristic search explorations, that is the constant size of the list $\mathcal{I}'$ of high-conflict species, can be justified; possibly the size of $\mathcal{I}'$ would need to be linear, or at least logarithmic in the number of species studied, if we did an "average-case" analysis of the method. "Easy alignments," for which the split with the largest weak weight is the correct one, do not require heuristic searches at all, and the more visible the shared novelties are, the easier the heuristic search is. In general, the amount of information in the alignment, the ability of our method to use this information, as well as the amount of misleading information in the alignment, and the power of our method not to be mislead, give rise to a lot of interesting theoretical issues.

A variant is to explore local optima as well as the global optimum found so far, by branching whenever the new-found split is an improvement to the current one, but not necessarily to the minimum-conflict one. Running time can be controlled further by fixing the number of branching events allowed. In the current implementation, a constant $maxforks = 4$ is the upper bound on the number of recursive calls, except for those starting with a *new* optimum G_{min}. In other words, the number of recursive calls due to the exploration of local optima and due to the exploration of trailing species in $\mathcal{I}'$ have a common upper bound.

3.14 MCOPE Running Time Analysis

In the following analysis of the overall time complexity of MCOPE, we ignore the non-essential consideration of neighbors (section 3.12), as well as the detection of runs in consecutive indices in the set of pattern-supporting columns (section 3.11). The latter calculation is only needed if columns with gaps are considered for the analysis, and an empirical comparison of running times with and without detection of runs reveals that it imposes only a neglegible performance penalty.

The two parameters of the analysis are the number of sequences m and the length of the alignment r. As we have seen, MCOPE estimates the tree top-down, in a divide-and-conquer fashion, employing heuristic searches for minimum conflict at each node. Basically, these heuristic searches require the calculation of the weak spectrum in time $O(mr)$ worst-case, and split conflict calculations.

Since divide-and-conquer is employed, calculations will actually be done with smaller and smaller sets of sequences; their size is denoted by the variable ℓ, and the size of the corresponding projection of the alignment is q. We have used these two parameters in the running time estimates that we calculated up to now for certain parts of the algorithm, culminating in the worst-case estimation of the time complexity of the conflict value of one split,

$$O(\frac{q^2}{\log q} \cdot \ell),$$

see section 3.9.

The list $\mathcal{I}'$ of high-conflict species to be moved has constant size, using a filter that truncates the list if the maximum number of elements f_{max} is exceeded. Then, the heuristic search via `movespecies` requires $O(1)$ split conflict calculations, since each recursive call to `movespecies` starts with one element less in the list of high-conflict species. As discussed before, such a truncation may reduce the accuracy of MCOPE, but complex issues may need to be resolved to understand its impact.

Since the heuristic search does $O(1)$ split conflict calculations, the number of split conflict calculations during the entire tree traversal of MCOPE-`Heuristic` is $O(m)$ in the case of a `caterpillar` tree as shown in Fig. 3.29, panel 1. (The internal nodes of this completely unbalanced tree are labeled with the number of species that are considered once they are visited, but the number of species does not affect the number of heuristic searches, which is constant. For comparison, the same labelling has been done for a balanced tree in panel 2. If the number of heuristic searches would depend on the number of species considered, the effect of the tree topology would be profound.)

Continuing the worst-case analysis, we instantiate ℓ by m and q by r, such that one split calculation now takes time $O(\frac{r^2}{\log r} \cdot m)$ worst-case. (In other words, we assume that the tree is not balanced, and that the projected alignments have no smaller length than the alignment used at the start. The latter assumption implies alignments composed only of very variable columns, and that the size of the alphabet is sufficiently large to allow such variation for the given number of species.)

Considering one calculation of the weak spectrum before each start of the heuristic search, in time $O(mr)$ worst-case, the overall running time of MCOPE as described is now

$$O(m) \cdot O(mr) + O(m) \cdot O(\frac{r^2}{\log r} \cdot m) = O(\frac{r^2}{\log r} \cdot m^2),$$

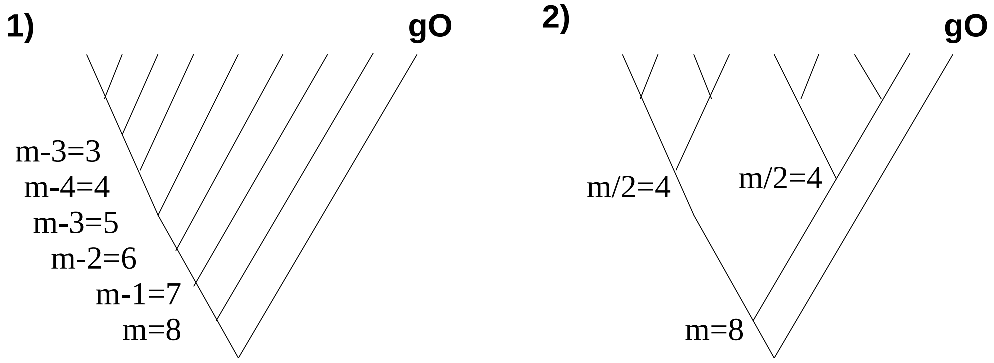
for an unbalanced tree with $O(m)$ interior nodes.

Assuming balanced trees, the number of weak spectrum and split conflict calculations during the tree traversal and the heuristic searches is $O(\log m)$, and the total running time would then be

$$O(\frac{r^2}{\log r} \cdot m \log m).$$

The main caveat regarding the preceding calculations is the complexity of the heuristic search, which may not be successful if the bound on the number of explorations is constant as assumed. Then again, it is likely that the amount of explorations necessary depends on the nature of the alignment, in particular on the amount of misleading information it includes. As noted, our theoretical understanding of MCOPE is still insufficient to tackle these issues.

Figure 3.29: Caterpillar tree, and balanced tree.



Chapter 4

Validation, Discussion and Future Work

4.1 Artificial versus Real Datasets

Intensive validation on both real and artificial data has been performed with very good results. These are presented in this chapter. Some intensive research, including the consultation of many people, gives the impression that there are currently no benchmark datasets for phylogeny estimation, either published in the literature, or on the World-Wide-Web. The tradeoff between

- unrealistic assumptions employed for the generation of artificial data, and
- the impossibility to know the true phylogeny for real data

poses major problems for benchmarking phylogeny estimation methods. Nevertheless, we can have confidence in the good performance of a method that is carefully validated with both kinds of data.

4.2 The MCOPE Standard Parameter Set

For both the artificial and real datasets used for validation, the following `standard parameter set` is used for all calculations.

- The *acceptable validity estimate* v_0 is set to 0.75. Validity estimates are activated symmetrically with a non-residual activation, using exponent $\eta = 5$, as described in section 3.2.3 starting on page 51.
- The *acceptable pattern count* s_0 is estimated from the data as described in section 3.7, depending on the regularity of the distribution of all pattern counts. Pattern counts are activated in a non-residual manner, again using exponent $\eta = 5$.

- The smoothness of the sigmoid function activating pattern counts depends on the acceptable pattern count, as explained in section 3.7.

- The *minimum column count* δ is set to be the log_2 of the number of variable alignment columns. This yields small values around 5-10 for the splits investigated; patterns supported by fewer columns are ignored and should not bear relevant conflict.

- The smoothness of the sigmoid functions operating on values in the range [0..1] is set to 0.1.

- The *minimum invariability threshold* τ is set to 0. In this way, a maximum of information is extracted from the alignment subcolumns by relative majority voting.

- The parameters for neighbor pattern consideration are set as follows:

  - The *acceptable hamming distance* μ_h between a pattern type and its neighbor is estimated from the data, considering the size of the subgroup of species under consideration, and the sparseness of the distribution of all patterns, cf. section 3.12.2.
  - The *neighbor impact factor* ω_ϵ is set to 0.25. Neighbors contribute at most 25% of their pattern count to the pattern under investigation.
  - The *minimum preliminary neighbor weight* ε is set to 0.5, on a scale of 0 to 1. If a neighbor has less preliminary weight based on its hamming distance, the softness of the symmetric difference, and the number of columns supporting it, it is not considered.
  - The slope smoothness of the activation for the matching rate congruence between neighbors is set to 0.25.

  Usually, results of MCOPE without neighbor consideration are only slightly worse.

- All columns with unknown nucleotides / missing data (usually coded "?" or "N" in the alignment) are removed.

- If columns with gaps are considered, runs in the indices of columns supporting a pattern are removed in part using standard parameters as described in section 3.11.

- For the real data, the parameters used for the heuristic search are the ones suggested in section 3.13; for the artificial data the exploration is reduced to "just moving high-conflict species", starting with the split of maximum weak weight.

4.3 Artificial datasets

For the generation of artificial datasets, the tool ROSE (Random Generation of Nucleotide Sequences) [32], Version 1.0.1, was used. Rose allows a wide array of parameters; we restrict our analysis to nucleotide sequences generated under the following setup:

- A random tree topology with 32 leaves is constructed by ROSE. The `mutability`, that is the percentage of nucleotides modified along one branch of the tree (also known as the "percent accepted mutations", or PAM) is set by editing the tree lengths, and feeding the tree back to ROSE, for sequence generation only. This manipulation is necessary since per default, ROSE 1.0.1 generates trees with a very unequal branch length distribution, if the PAM-value (expressed in terms of "average distance") is supplied directly. (Basically, ROSE 1.0.1 obtains trees by randomly dropping sequences and corresponding branches from a fully resolved binary tree with approx. 1000 nodes and all-equal user-specified branch lengths. Imagine a path from the root to a surviving leaf. Since the more outgoing branches are deleted, the closer one gets to the leaf, internal branchlengths close to the root are a lot smaller than the ones close to the leaf.)

- 32 sequences are generated as described, with average sequence lengths of 500, 1000 and 1500 nucleotides. We then split the tree at the root, and the first subtree contributes the sequences to be analyzed, whereas the second subtree contributes the outgroup by relative majority rule. This mechanism results in trees of different size and topology; the average size is 16 sequences.

- The following simple substitution probability matrix is used. Nucleotides are substituted with 1% probability per unit of branchlength (1 PAM), and all three kinds of substitutions are given the same probability of 0.003333.

$$\begin{aligned} M &= \begin{pmatrix} m_{\mathtt{A}\to\mathtt{A}} & m_{\mathtt{A}\to\mathtt{C}} & m_{\mathtt{A}\to\mathtt{G}} & m_{\mathtt{A}\to\mathtt{T}} \\ m_{\mathtt{C}\to\mathtt{A}} & m_{\mathtt{C}\to\mathtt{C}} & m_{\mathtt{C}\to\mathtt{G}} & m_{\mathtt{C}\to\mathtt{T}} \\ m_{\mathtt{G}\to\mathtt{A}} & m_{\mathtt{G}\to\mathtt{C}} & m_{\mathtt{G}\to\mathtt{G}} & m_{\mathtt{G}\to\mathtt{T}} \\ m_{\mathtt{T}\to\mathtt{A}} & m_{\mathtt{T}\to\mathtt{C}} & m_{\mathtt{T}\to\mathtt{G}} & m_{\mathtt{T}\to\mathtt{T}} \end{pmatrix} \\ &= \begin{pmatrix} 0.99 & 0.003333 & 0.003333 & 0.003333 \\ 0.003333 & 0.99 & 0.003333 & 0.003333 \\ 0.003333 & 0.003333 & 0.99 & 0.003333 \\ 0.003333 & 0.003333 & 0.003333 & 0.99 \end{pmatrix}. \end{aligned}$$

- In ROSE, insertions and deletions depend on the "average distance" value (called "Relatedness" in the ROSE 1.0.1 interface), user-supplied "thresholds", and user-supplied indel length functions. As described in [32], the PAM-value is multiplied with the depth of the tree, yielding the "average distance". The insertion and deletion thresholds are both set to 0.1, and the length of both insertions and deletions follow the following distribution:

$$frequency(length) = 1/(2 \cdot 1.5^{length}),$$

not considering lengths larger than 10. The following table lists the length distribution explicitly; note that indels larger than 10 may appear in the sequences due to the multiple indels.

length	1	2	3	4	5	6	7	8	9	10
frequency	0.333	0.222	0.148	0.099	0.066	0.044	0.029	0.020	0.013	0.009

- We do *not* use the following possibilities offered by ROSE:
  - Sequence motifs, where substitution probabilities differ across the sequence length.
  - Definition of a non-uniform distribution to be used for creating the sequence at the root of the tree, and insertions.
  - Selection of internal node sequences for inclusion into the alignment to be analyzed.

For most of the data shown, 32 independently created phylogenies were analyzed using minimum conflict. In other words, 32 `runs` with the same average number of sequences (i.e. approx. 16), the same average sequence length (i.e. 500, 1000 or 1500 nucleotides), and the same mutability were performed for each data point. In a few cases, execution of the software was terminated prematurely due to external factors; in any case, at least 30 phylogenies were created. The heuristic search relied entirely on the `movespecies` heuristic starting with the split of strongest weak weight; no spectra were obtained and filtered.

For each run, the `error count` was calculated as the number of incorrectly established splits across the whole tree. Whenever an incorrect split was favored by MCOPE, the error count was incremented and the calculation was resumed with the correct split as if nothing happened. The main alternative to this "count of incorrectly estimated splits" would have been the completion of a run without any restart, and the comparison of the resulting tree with the true tree, e.g. by establishing the number of false positive and false negative bipartitions in the estimated tree. (In particular, the "Robinson-Foulds score" combines both numbers into one estimate, see [44].) The advantages and drawbacks of our "count of incorrectly estimated splits", in relation to the more direct comparison of the calculated and the correct tree, are as follows.

- Our error count may overreport errors because the same species or set of species can be mishandled several times. For example, a long-branch species may be split off incorrectly at the root of the tree, but the correct tree then forces the species back into one of the two correct subtrees. Following the resumption of calculations with the two correct subtrees, the species is mishandled again in one of the subtrees, etc, etc.
- Underreporting of errors is possible because without the resumptions with the correct split, our tree estimation method may make many more errors

> in subsequent divide-and-conquer steps. However, we can view a resumption as a new run with fewer sequences, for which we can, in an ad-hoc manner, assume at least a similar error probability as the one obtained for the larger set of sequences. In essence, our conjecture is that the error probability does not feature any significant change from top to bottom : it does not depend much on the number of sequences analyzed.

In any case, the differences between any reasonably defined error counts vanish as the counts approach zero, so we do not run into interpretation problems if our error count is close to zero.

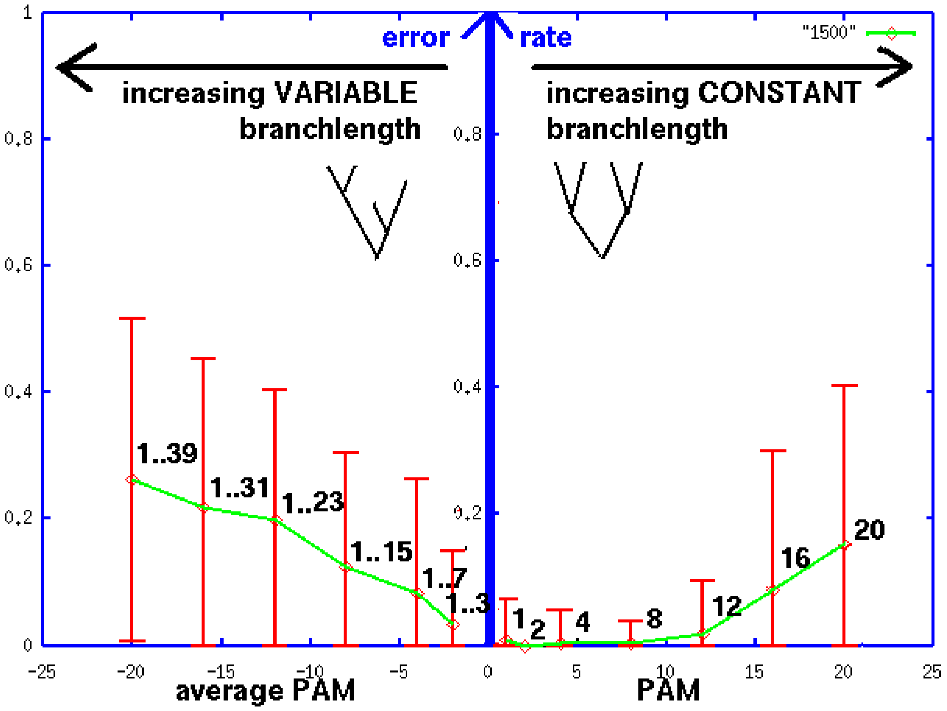
The `error rate` of a single run is defined as the relative frequency of error, i.e. the error count is divided by the number of splits to be estimated for the tree under consideration. The `error rate` of a set of runs is the average error rate taken over all runs performed.

Results obtained by inspecting random trees with approximately 1500, 1000 and 500 nucleotides are shown in the following set of figures, Figs. 4.1, 4.2 and 4.3. In all 3 figures, the vertical axis is labelled with the average error rate established over at least 30 runs, and the horizontal axis is labelled with the specifics of the run. This PAM-value, or mutability, is the number of applications of the substitution probability matrix along *one* branch of the artificial tree, from the ancestral to the descendant node. Naturally, the percentage of substitutions introduced along a path of branches is much larger. For example, the percentage of substitutions between two sister group sequences is almost twice the mutability, unless multiple hits cause saturation effects in case of very large mutabilities. For the right-hand side of the figures, the mutability is the same for each branch, while on the left-hand side, it varies as indicated, and the axis refers to the *average* mutability, marked as such by the minus ("-") sign.

The figures include so-called "error" bars for purposes of decoration only, based on an unsound attempt to calculate a standard "error" of the mean as a basis for 95% confidence intervals. (Note that "standard error of the mean" is a technical statistical term, referring to the *"error"* that may be attached to the mean value displayed. Confusion may arise because our variable is an *error* rate. However, "error" in the statistical sense is always put into quotation marks in this chapter, not just to avoid confusion, but also because it is not calculated in a statistically sound manner. In summary, that ""error"" in doing statistics makes it easy to distinguish between our error variable and the technical "error" term in statistics.)

Our calculation of "error" bars is unsound because error rates do not feature a symmetric distribution around the mean value – there are no negative rates ! Furthermore, the "error" bars may be inflated because they are based on a distribution of discrete values; if we investigate a tree with 10 internal nodes, 10 splits need to be estimated, and the error can be 0, 0.1, 0.2, 0.3, etc, but not 0.25. We also note that 32 runs are just enough to calculate "error" bars; the author of [34] writes that "With small samples – say under 30 observations – larger multiples of the standard error are needed to set confidence limits." (For the record, we use ± 2 standard deviations, aiming at a 95% confidence

Figure 4.1: Error rates for artificial data, 1500 nucleotides.

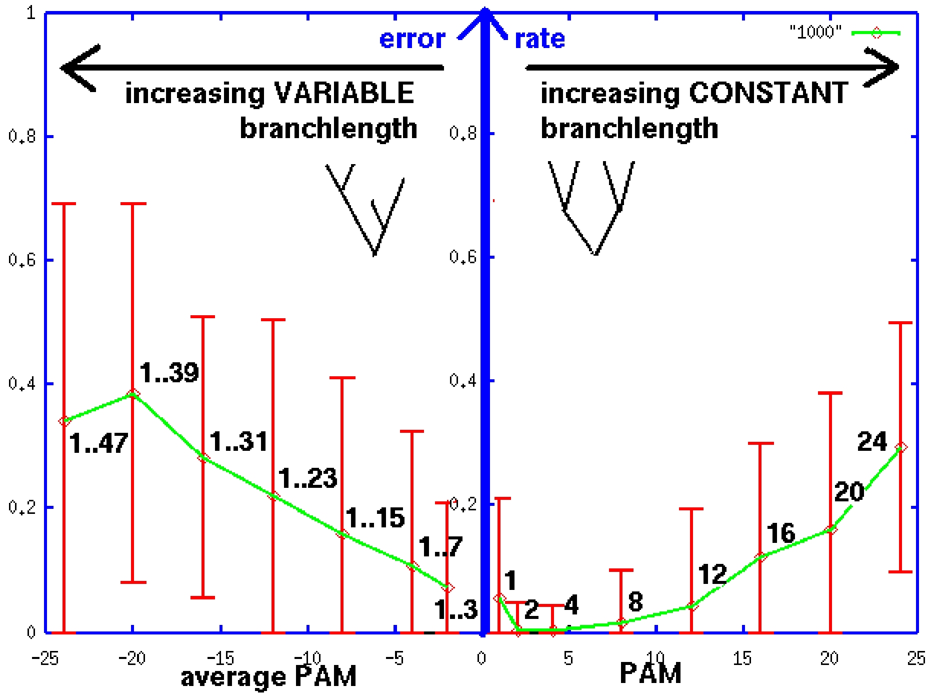


interval.)

Using approx. 16 sequences with 1500 nucleotides, 32 independently created phylogenies established an error rate of zero or almost zero given a mutability of 1 - 12 PAM (Fig. 4.1, right). Larger mutability triggers more and more error, like 8% for 16 PAM, and 17% for 20 PAM. If the mutability varies randomly between 1 and 3 PAM on different branches of the tree, we observe about 3% error. Again, larger branchlengths result in more and more error (Fig. 4.1, left). To some degree, the large error rates to the left can be expected; after all, mutability variation between 1 and 48 PAM implies almost complete randomizations taking place across some branches of the tree, and randomization effects are even much stronger along paths of the tree.

Fig. 4.2, right, displays similar results for an average sequence length of 1000 nucleotides. We observe an average error rate of zero or almost zero given a mutability of 2 - 8 PAM. A smaller mutability of 1 PAM renders the alignment less informative, so we can indeed expect a higher error rate. Again, larger mutability triggers more and more error (Fig. 4.2, right). If the mutability

Figure 4.2: Error rates for artificial data, 1000 nucleotides.



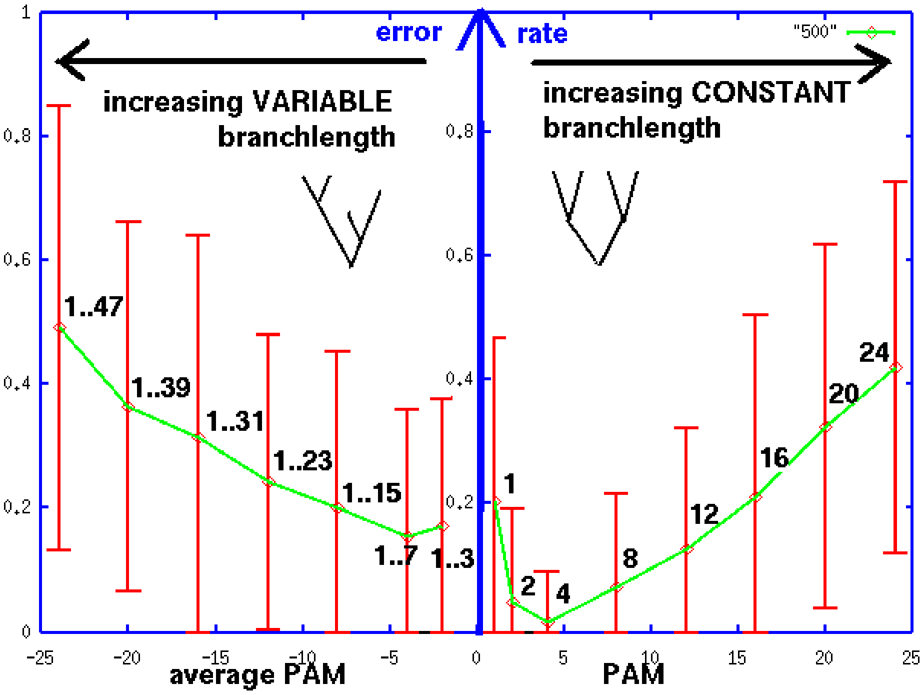
varies randomly between 1 and 3 PAM on different branches of the tree, about 7% error results, and for variation between 1 and 7 PAM, we observe about 10% error. Again, larger branchlengths result in more and more error (Fig. 4.2, left).

For sequences with only 500 nucleotides, we note much higher error rates (Fig. 4.3, right), unless mutability is around 4 PAM. In particular, the rise in error for small mutability is more pronounced. Moreover, in these cases variation of branchlength within the trees studied increases the error rate quite a lot (Fig. 4.3, left). For example, we record about 16% incorrectly estimated bipartitions in the case of variation between 1 and 7 PAM. However, for half of these errors the correct bipartition has a conflict value very close to the erroneous one (data not shown), and polytomies should have been flagged.

The following problems are not handled well by MCOPE, analyzing the artificial datasets.

- Sometimes, the outgroup is too close to the group under investigation. Then, a very high standard matching rate results, which is not exceeded by the observed matching rate because of sampling error, even though the

Figure 4.3: Error rates for artificial data, 500 nucleotides.



pattern is due to erosion.

- Rarely, high impact of neighbors on pattern counts lifts the acceptable pattern count threshold such that valid patterns do no longer qualify if they have few neighbors. In such cases a general cutoff on the impact of neighbors may be helpful. However, the natural way to reduce their impact is to increase the quality criteria they need to meet, like softness of the symmetric difference. High-quality neighbors should be considered no matter what their weighted count is, possibly even if it exceeds the pattern count of the original pattern.

Current knowledge about the relationship between the process used for the generation of the artificial data, and the evolutionary processes that are behind the real data, is very limited. Calculating mutabilities / nucleotide substitution rates from real data is not straightforward, as exemplified by the complex procedures used in [39, 40]. Therefore, some intensive research is needed to find good estimates for the mutabilities that we can expect in real data, not to mention their variation across the different branches, and we should not derive any detailed conclusions yet based on the performance of MCOPE on artificial data.

4.4 Real-life datasets

Artificial datasets are an unsatisfactory substitute for biological data; the model needed to generate the data must naturally be inadequate. Evolutionary processes do not fit into a simple model, and complex models usually imply many false assumptions. Moreover, systematic errors in phylogeny estimation, including branch attraction phenomena, are hard to come by except for a careful study of real data.

We have selected real data from two sources. We will reinvestigate published studies, and we will assemble datasets from an alignment database. We will take great care that the latter are assembled in an almost objective manner, and not in a manner that suits us best.

In the case of the published studies, we conjecture that current methods have fallen victim to erosion whenever MCOPE estimates the presumably correct split not estimated by the authors. In the first study, to be presented in section 4.4.1, the incorrect split has high bootstrap support. However, bootstrap values (see e.g. [36]) may be deceived by systematic error – e.g. these indicate maximum support (100%) for any data if the method just builds up a caterpillar tree of the sequences in input order; resampling will yield such an artifact tree every time.

All real-life datasets investigated are collections of 18S-rDNA, with two exceptions.

- The Arthropod dataset (section 4.4.2) includes both 18S-rDNA and 28S-rDNA. 28S-rRNA is also a component of the ribosomes of a cell.
- The Mammalia dataset (section 4.4.5) features mitochondrial DNA.

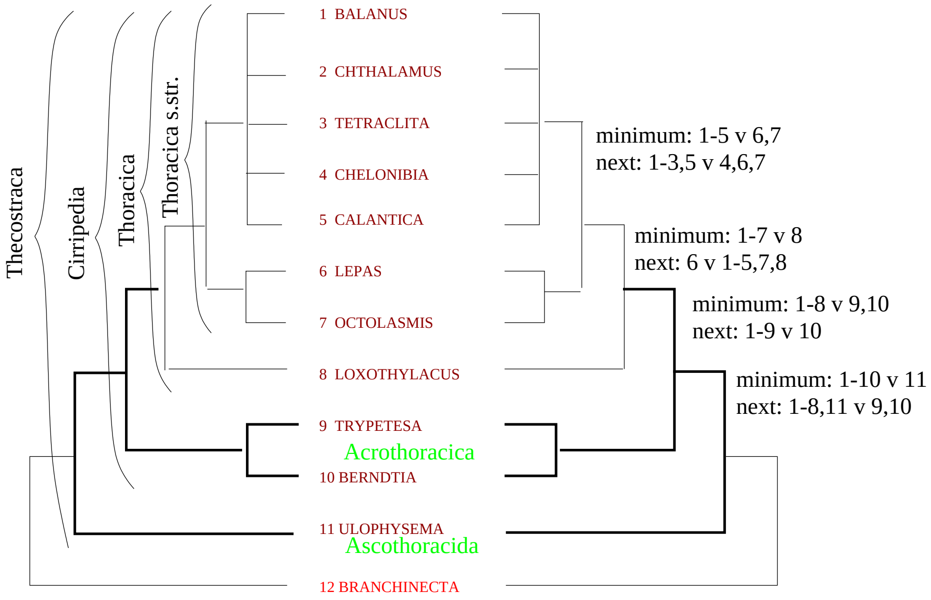
In general, rDNA is a very popular gene for phylogeny reconstruction since its product has a unique function within the cell, and it has many conserved regions that can be aligned, often based on structural data. Moreover, the conserved regions enable the use of standard sequencing primers. Also, rDNA occurs in many repeats, which makes it even more easier to sequence.

4.4.1 Analysis of the Crustacea Dataset

The Crustacea dataset published by Spears *et al* [31] has been used as our running example. Alignments and sample calculations shown in previous chapters were based on it.

The data comprise 18S-rDNA from twelve species, one species (Branchinecta, 12) from the Branchiopoda group, and 11 species from their putative sister group Thecostraca. The Thecostraca split into Cirripedia and Ascothoracica,

Figure 4.4: Putative correct tree (left) and minimum conflict tree (right) for the Crustacea dataset.

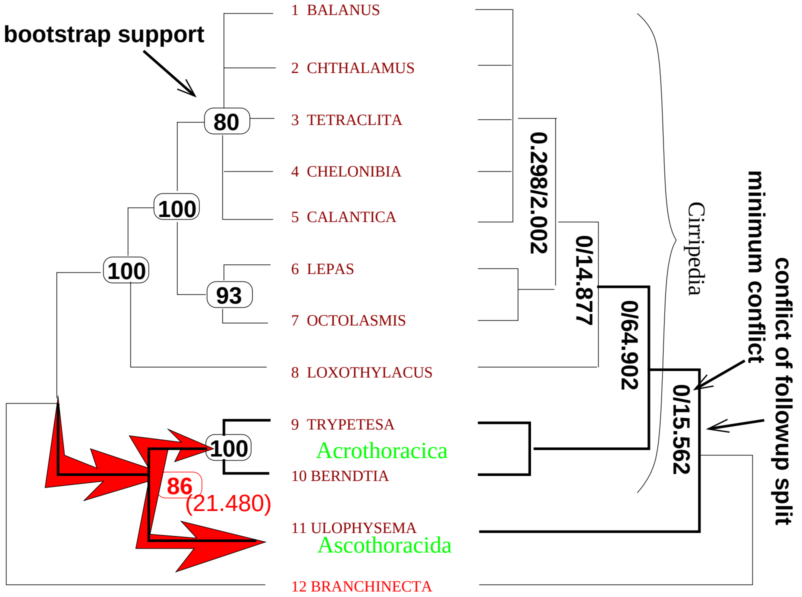


and the Cirripedia split into Acrothoracida and Thoracica. Thoracica in turn are comprised of Rhizocephala and Thoracica sensu stricto. The tree from Fig. 4.4, left, is usually assumed to be correct, based on morphological data [31].

The next figure, Fig. 4.5, left, features the tree obtained by Spears *et al* [31]. The authors comment their tree as follows: "Parsimony, invariants and neighbor-joining analyses all showed the Ascothoracida and Acrothoracica to be sister taxa [...] Although we certainly do not reject the considerable molecular data supporting a close relationship between the Acrothoracica and Ascothoracica, we suggest that the Acrothoracica diverged very early from the cirripedian lineage [...]". (The method of "Invariants" goes back to [17, 2], and "neighbor-joining" [28] refers to a popular distance method. Both methods are also explained in [36].)

As we have shown, the published tree is very likely due to erosion, and the more plausible tree featuring the Cirripedia (species 1-10) as a monophylum is clearly found by minimum conflict (Fig. 4.4 and 4.5, right.) In Fig. 4.4, the split of minimum conflict used to draw the tree is also listed as the first label ("minimum"). The follow-up split, that is the split with the second-lowest conflict, is listed as well ("next"). In Fig. 4.5, the label attached to an internal node of the minimum conflict tree lists the minimum conflict value established, followed by the conflict of the second-best split. Furthermore, on the left of the figure, the conflict value obtained for the split featured by the published tree (21.480) is noted next to its bootstrap value. In other words, the split 1-10 v 11

Figure 4.5: Published tree, redrawn (left) and minimum conflict tree (right) for the Crustacea dataset.

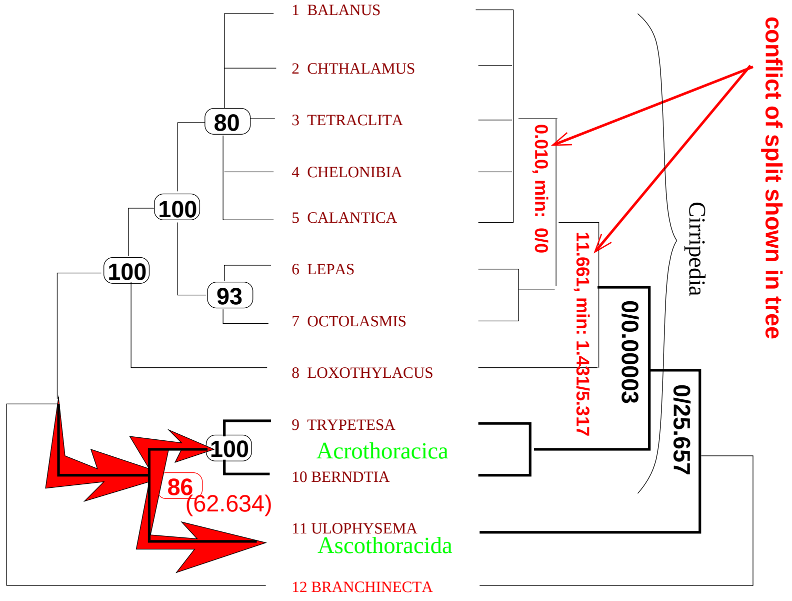


triggers zero conflict, followed by the split 1-8,11 v 9,10, which has a conflict of 15.562. The split 1-8 v 9-11 triggers a conflict value of 21.480. We obtain this minimum conflict tree with the standard parameter set outlined in section 4.2, ignoring the parts of the alignment featuring gaps.

Using alignment columns with gaps as well, we estimate the tree in Fig. 4.6, right, but only the first two splits. Again, 1-10 v 11 has zero conflict, followed up by 1-8,11 v 9,10 (conflict 25.657). The incorrect split 1-8 v 9-11 has conflict 62.634. The next correct split 1-8 v 9,10 is recovered, but not too well; split 1-9 v 10 has conflict 0.00003 due to insufficient validity of the pattern ”9” in 1-9. Further correct splits are not recovered at all. It is possible that our gap-handling mechanism does not cope well with the Crustacea dataset, and/or the gaps in this dataset are misleading anyway. Ignoring neighbors of patterns and alignment columns with gaps, we estimate the tree in Fig. 4.7, right, where conflict values are almost identical to the ones estimated with neighbor consideration. Ignoring neighbors of patterns but not alignment columns with gaps, we estimate the same tree as in Fig. 4.6, right, with very similar conflict values; for the Crustacea dataset, neighbors do not contribute much information. All these trees are based on calculations like the ones presented in our running example, which uses the first 50 variable columns without gaps or unknown nucleotides / missing data.

We now provide some more details on the heuristic search leading to the minimum-conflict tree for the Crustacea dataset, considering neighbors and ignoring alignment columns with gaps (Fig. 4.5, right).

Figure 4.6: Published tree, redrawn (left) and conflict tree considering gaps (right) for the Crustacea dataset.



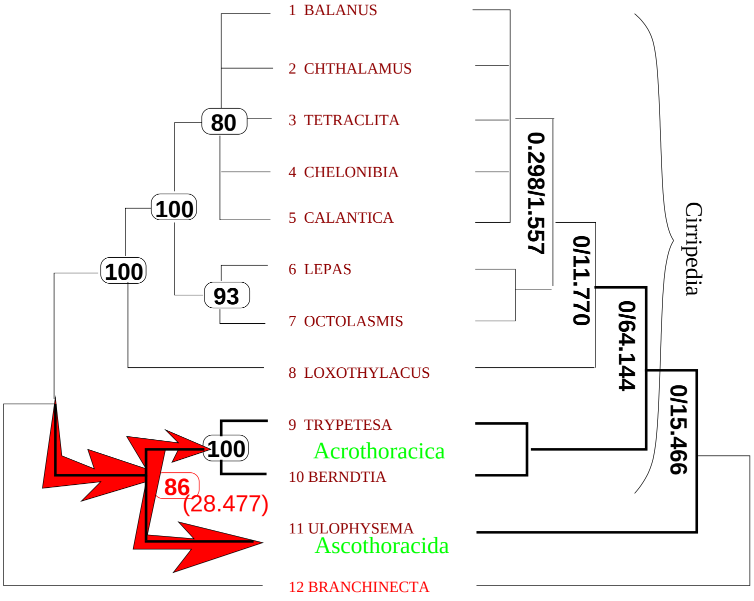
Using species 12 as the outgroup, we first obtain the following heuristic split search results:

Split			Conflict	Weak Weight
1-10	v	11	0.000	75
9,10	v	1-8,11	15.562	49
1-8	v	9-11	21.450	226
10	v	1-9,11	34.356	8
9	v	1-8,10,11	34.356	5
8	v	1-7,9-11	117.323	90
6	v	1-5,7-11	123.932	12
7	v	1-6,8-11	124.563	16
1-7	v	8-11	125.547	31

The search starts with the calculation of the weak weights, now listed in the last column. The weak spectrum starts with the split 1-8 v 9-11, for which weak weight is provided by 226 columns, followed by 8 v 1-7,9-11, 1-10 v 11, 9,10 v 1-8,11, 1-7 v 8-11, etc. Sorting the weak spectrum by split conflict almost results in the table shown; the heuristic search encounters only one further split, namely 9 v 1-8,10,11, by moving species 10 out of 9,10 v 1-8,11. Since the number of variable columns is 298 for the alignment of species 1-11, the minimum column count for the preceding conflict calculations is $\lceil log_2 298 \rceil = 8$.

The outgroup for the species 1-10 is 12, repeating the calculation done in section 3.10, for the full alignment. The following search results are then obtained

Figure 4.7: Published tree, redrawn (left) and minimum conflict tree ignoring neighbors (right) for the Crustacea dataset.



for species 1-10 with outgroup 12:

Split			Conflict	Weak Weight
1-8	v	9,10	0.000	275
1-9	v	10	64.902	8
9	v	1-8,10	77.390	5
8	v	1-7,9,10	132.214	96
6	v	1-5,7-10	137.972	14
4	v	1-3,5-10	138.388	8
7	v	1-6,8-10	138.710	16
5	v	1-4,6-10	138.819	6
2	v	1,3-10	138.907	15
1-7	v	8-10	144.000	35

This time, the minimum-conflict split is also the one with the maximum weak weight. This will also hold for the investigation of species 1-8, but not for the investigation of species 1-7, where the splits 1-6 v 7 and 1-5,7 v 6 have maximum weak weight (data not shown). The outgroup selected for 1-8 is again species 12, but for 1-7, species 8 triggers a smaller spread of matching rates, and is therefore elected as outgroup.

Since the minimum column count is the base-2 logarithm of the number of variable columns, it gradually decreases with this number, which in turn decreases due to the loss of variation in the species that are no longer considered.

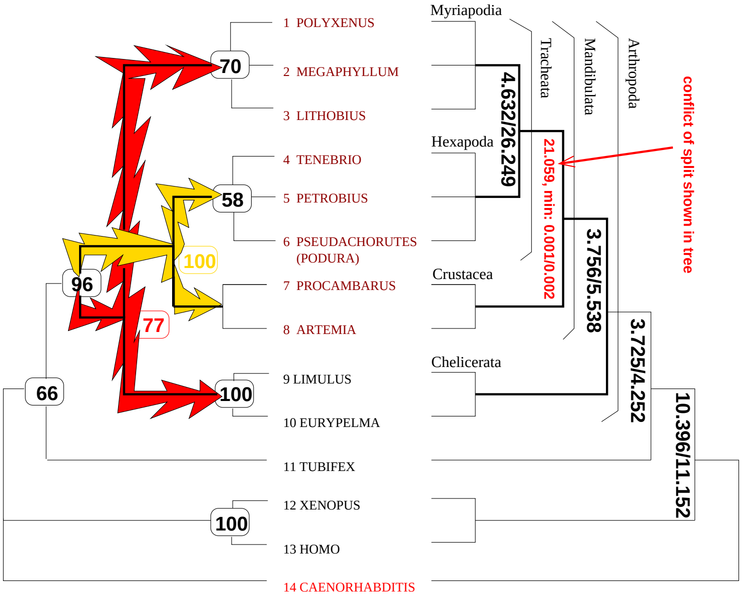
In this case, the projected alignments featuring species 1-10, 1-8 and 1-7 have 267, 137 and 75 variable columns, respectively, and the minimum column counts are 8, 7 and 6.

The calculation of the Crustacea minimum conflict tree then continues with an insufficient number of just 45 variable columns.

4.4.2 Analysis of the Arthropod Dataset

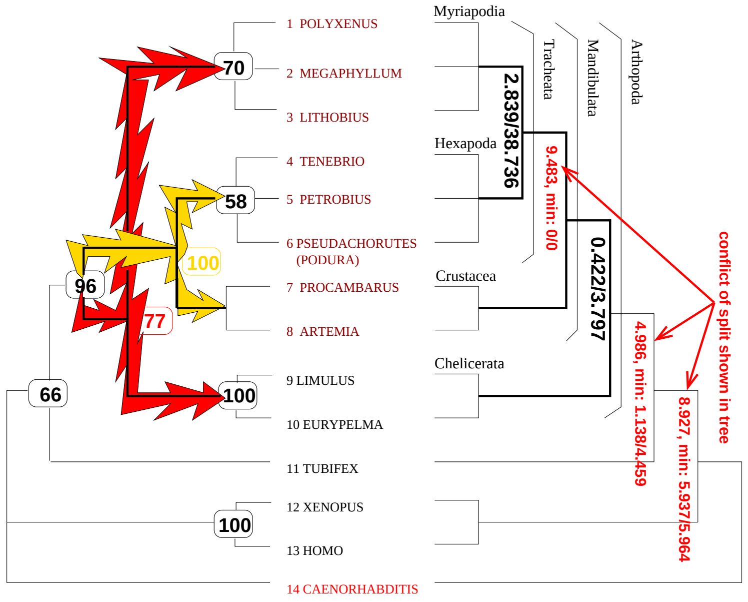
The publication of the analysis of the Arthropod dataset, an alignment of parts of both 18S-rDNA and 28S-rDNA, by Friedrich and Tautz [6] led to significant debate. Analyzing arthropod, vertebrate, annelid and nematode sequences, it suggests that the Mandibulata, species 1-8, do not comprise a monophyletic group, nor do the Tracheata, species 1-6 (see Fig. 4.8, left. Bootstrap support is again given in boxes. The tree published also indicates branchlengths, and resolves the two polytomies in the same way as MCOPE).

Figure 4.8: Published tree, redrawn (left) and conflict tree (right), for the Arthropod dataset.



The tree is estimated by maximum likelihood, and the authors state that bootstrap analysis with maximum likelihood as well as parsimony and neighbor joining ([28], a distance method) give trees that do not differ substantially (see [6] for details.) Using the alignment that was distributed by the authors, where the species ”Pseudachorutes” was replaced by ”Podura”, MCOPE is able to recover monophyly of the Mandibulata, despite bootstrap support of 77% in favor of non-monophyletic Mandibulata (see Fig. 4.8, right. As before, a split in the conflict tree is usually labeled by its minimum conflict value, and the

Figure 4.9: Published tree, redrawn (left) and conflict tree ignoring neighbors (right), for the Arthropod dataset.



conflict value of the next-best split. If the minimum-conflict split is not the presumably correct split, the conflict value of the latter is listed first, followed by the minimum conflict value and the second-best value. This case is indicated in red).

The reason for the putative incorrect split is once again deemed to be erosion: Crustacea and Hexapoda form one group in the Friedrich and Tautz publication because they gained substitutions fast, while Myriapoda and Chelicerata share old character states. Monophyly of the Tracheata cannot be established; the minimum-conflict split for 1-8 is 3 v 1,2-8 (conflict 0.001), followed by 1 v 2-8 (conflict 0.002) and 2 v 1,3-8 (conflict 0.013). The split 1-6 v 7,8 has conflict 21.059.

There are just six gap characters in the alignment provided. However, we can investigate the effect of the neighbors (Fig. 4.9). Without neighbors, the incorrect split 2 v 1,3-13 has a minimum conflict of 5.937, closely followed up by 1 v 1-13 and the split 1-10,12,13 v 11. The correct split 1-11 v 12,13 has a conflict of 8.926. The correct split of 1-11, i.e. 1-10 v 11, cannot be found either, favoring 9,10 v 1-8,11 instead. For species 1-10, the correct split 9,10 v 1-8 is recovered. Monophyletic Tracheata are again not supported.

The number of variable characters in the alignment considered by MCOPE is rather constant; it goes down from 695 sites for group 1-13 to 645 sites for group 1-11, and then down to 610 sites for group 1-10. Even for group 1-6, 461 variable sites are still available, and we conclude that there is a lot variability in the data. Nevertheless, MCOPE seems to be able to make some sense out of this noisy dataset, in particular if neighbors are considered.

4.4.3 Analysis of the Bilateria Dataset

Up to now, we have reanalyzed datasets provided by other authors. For these datasets, we have no leverage in the choice of species, and the alignment is given by a black box (the authors used some computerized method and then corrected the alignment "by eye"). This time, we have constructed our own dataset from the alignment of the RDP (Ribosomal Database Project, [19]) database, which is guided by structural information. It is possible that this alignment is more objective than the alignments considered so far, since it is not corrected by eye, for a particular study.

The more important point however is the choice of species – how can it be done in an objective way, and still yield a set of species for which an "undisputed" correct tree can be constructed ? The RDP database offers a sequence query facility (the "Phylogenetic Tree Browser") that has a crude phylogenetic organization which we will finetune a bit, and then we can run the following procedure to obtain a selection of species that is both objective and classifiable to a very high degree.

1. Take a group of species to be studied.
2. Select the outgroup such that it is the first group of species listed on the same level. (If the first group in the list is the group under study, select the second one.)
3. Explore the selected groups. On the level of the group selected, always take *the first two* groups of species, and explore these in the same way.
4. Once you reach the species level, take *the first two* species.

The "take *the first two*" rule renders our selection process as objective as possible; *"the first two"* groups are predefined by the browser, and *"the first two"* species are predefined by the order in which groups and species are listed by the browser. The "take *the first two*" rule also helps us to select species such that the tree is "almost" undisputed; adding a third group would imply that a debate is possible on the correct classification of the three groups. (Our rule does not select the most "representative" groups or species; "representative" is a subjective criterion that is sacrificed in favor of a strict rule that just uses the rather arbitrary order in the listings given to us. We note that usually, reconstructing phylogenies becomes easier if "representative" species are used for the various groups.)

As of June 2000, the RDP tree is organized as in the table below; we have expanded the groups by following the procedure just outlined, and use boldface to highlight the species selected accordingly. The dots (....) represent groups of species that need not be considered, because they are not *the first two* groups, considering the finetuned phylogeny.

We have finetuned the RDP phylogeny by postulating the following:

- "(3.10.5) ECHINODERMATA_AND_HEMICHORDATES" and "(3.10.6) CHORDATA" are in one subtree. This subtree is called "Deuterostomia". Few biologists will doubt the resulting assumption that the Echinodermata and Chordata are in one subtree, and the Mollusca are in the other subtree.
- "Deuterostomia" and "(3.10.7) MOLLUSCA_AND_OTHERS" are in one subtree, to the exclusion of the other species in (3.10.1) – (3.10.4). This "undisputed" subtree is called "Bilateria".

(3.10) METAZOA_AND_RELATIVES
(3.10.1) CHOANOFLAGELLIDA
Drcy.salmo Dermocystidium salmonis (protist; fungi/metazoan incertae sedis)
Drcy.spSal Dermocystidium sp. (protist; fungi/metazoan incertae sedis)
....
(3.10.2) PORIFERA_AND_CTENOPHORA
(3.10.3) CNIDARIA_AND_PLACOZOANS
(3.10.4) NEMATODA_(ROUNDWORMS)

Bilateria: (3.10.5) + (3.10.6) + (3.10.7)
Deuterostomia: (3.10.5) + (3.10.6)

**(3.10.5) ECHINODERMATA_AND_HEMICHORDATES
**(3.10.5.1) ECHINODERMATA
**(3.10.5.1.1) ECHINOIDEA_(SEA_URCHINS)
Flls.zelan Fellaster zelandiae (sea urchin)
Enco.aberr Encope aberrans (sea urchin)
....
**(3.10.5.1.2) OPHIUROIDEA_(BRITTLE_STARS)
Ophp.japon Ophioplocus japonicus (brittle star)
Opph.acule Ophiopholis aculeata (brittle star)
....
....
***(3.10.5.2) ENTEROPNEUSTS_(ACORN_WORMS) – this group may be debated*
**(3.10.6) CHORDATA
**(3.10.6.1) GNATHOSTOMATA_(JAWED_VERTEBRATES)
**(3.10.6.1.1) AVES_(BIRDS)
Lthx.lutea Leiothrix lutea (red-billed leiothrix bird) – *fragment only*
Trgd.trgld Troglodytes troglodytes (wren) – *fragment only*
....
Trdu.migrt Turdus migratorius (thrush bird, unspecified)
....
Gall.gallu Gallus gallus (chicken)
....
**(3.10.6.1.2) MAMMALIA
Oryc.cunic Oryctolagus cuniculus str. New Zealand (rabbit) – *the position of this species in the tree is debated*
Oryc.cuni2 Oryctolagus cuniculus (rabbit)
Homo_sapi2 Homo sapiens (human)
Homo_sapie Homo sapiens (human)
Homo_sapi5 Homo sapiens (human)
Homo_sapi3 Homo sapiens (human)
Ratt.norw4 Rattus norvegicus (brown, common or Norway rat)
Ratt.norwe Rattus norvegicus (brown, common or Norway rat)
Ratt.norw2 Rattus norvegicus str. Sprague Dawley (brown, ... rat)
Mus_muscu4 Mus musculus (house mouse)
Ratt.norw3 Rattus norvegicus (brown, common or Norway rat)
Mus_muscul Mus musculus (common or house mouse)
Mus_muscu2 Mus musculus (house mouse)
....
**(3.10.6.2) AGNATHA_(JAWLESS_VERTEBRATES)

**(3.10.6.2.1) PETROMYZONTIFORMES_(LAMPREYS)

Lptr.aepyp Lampetra aepyptera (least brook lamprey)

Ptrm.marin Petromyzon marinus (sea lamprey; marine lamprey)

***(3.10.6.2.2) MYXINIFORMES_(HAGFISH) – this group may be debated*

....

*(3.10.7) MOLLUSCA_AND_OTHERS

*(3.10.7.1) MACTRIDAE_MOLLUSCA_(SURF_CLAMS)

Tres.nutta Tresus nuttali (surf clam)

Tres.capax Tresus capax (surf clam)

....

**(3.10.7.2) BRYOZOA_GROUP_I – this group is ill-defined*

*(3.10.7.3) OSTREIDAE_MOLLUSCA_(OYSTERS)

Otre.eduli Ostrea edulis (oyster)

Crst.virgi Crassostrea virginica (eastern oyster)

....

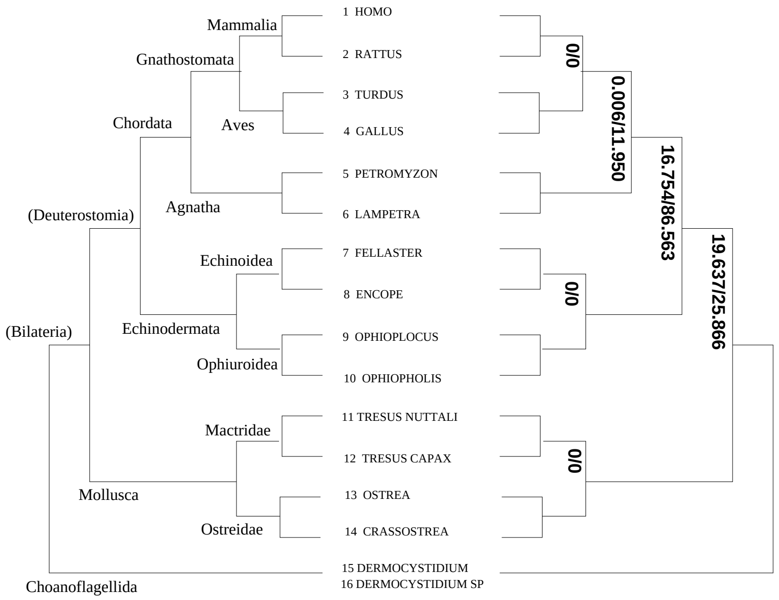
....

Now we can outline the strict process of species selection:

- The group to be studied is "Bilateria", (3.10.5) – (3.10.7).
- The first group on the same level is "(3.10.1) CHOANOFLAGELLIDA". It is used as the outgroup. Since we are already investigating two groups on this level of the finetuned phylogeny ("Bilateria", (3.10.5) – (3.10.7), and "(3.10.1) CHOANOFLAGELLIDA"), no further groups from this level are selected.
  - One level further, we investigate the "Bilateria". The first group is the "Deuterostomia", the next one is "(3.10.7) MOLLUSCA_AND_OTHERS", and we do not consider other groups on the same level.
    - One level further, we investigate the "Deuterostomia". The first group is "(3.10.5) ECHINODERMATA_AND_HEMICHORDATES", and the second one is "(3.10.6) CHORDATA". Within (3.10.5), we ignore the "unsafe" group "(3.10.5.2) ENTEROPNEUSTS_(ACORN_WORMS)", and only tackle "(3.10.5.1) ECHINODERMATA". From this group, we take the first two species of the first two subgroups.
    - On the same level, we investigate "(3.10.6) CHORDATA". The first subgroup is "(3.10.6.1) GNATHOSTOMATA_(JAWED_VERTEBRATES)", and the second one is "(3.10.6.2) AGNATHA_(JAWLESS_VERTEBRATES)". From the first subgroup, we take two birds and two mammals, since "(3.10.6.1.1) AVES_(BIRDS)" and "(3.10.6.1.2) MAMMALIA" are listed first. The two birds selected are the only ones for which not just an RNA fragment is available. Mammal selection ignores "duplicates" as well as species of debated phylogenetic origin (Oryctolagus). From the second subgroup, we take two species from "(3.10.6.2.1) PETROMYZONTIFORMES_(LAMPREYS)", ignoring "(3.10.6.2.2) MYXINIFORMES_(HAGFISH)" since its phylogenetic origin is also debated.
  - From "(3.10.7) MOLLUSCA_AND_OTHERS", we take the first two species of the first two subgroups, ignoring an ill-defined group "BRYOZOA_GROUP_I".

The result of this strict species selection process is the tree in Fig. 4.10 on the left, which is, moreover, hard to dispute. The minimum conflict tree on the

Figure 4.10: "Undisputed" tree (left) and minimum conflict tree (right) for the Bilateria dataset.



right recovers the phylogeny correctly, even though resolution deteriorates for the most internal nodes. The reason is a lack of variable columns – after all, we ignore all the columns that include unknown nuclotides / missing data, as specified by the MCOPE standard parameter set. (It is future work to rectify this necessity, by handling these cases in a reasonable way. Then again, permitting missing data may introduce subtle artifacts if e.g. sequences "from one study only" carry them. If the study concerns just molluscs, molluscs are "marked" by the presence of missing data, and subtle effects may drive them into one group just because of this.)

MCOPE starts off with 522 variable characters in the alignment of species 1-14, continues with 408 variable sites in the alignment of species 1-10, and can still resolve the correct split of 1-6, given 206 variable sites. Lists of many zero-conflict splits result for the remaining subtrees of species 1-4, 4-8 and 9-12, featuring 135, 177 and 191 variable characters, respectively.

Ignoring gap character states, MCOPE does one mistake, suggesting an incorrect subtree of Chordata and Mollusca, with the Echinodermata as the incorrect sister group. The corresponding split, 7-10 v 1-6,11-14, has conflict 16.812, but the second-best split with conflict 18.616 is the correct split 1-10 v 11-14.

4.4.4 Analysis of the Gnathostomata Dataset

MCOPE performance is mixed on the second dataset retrieved from the RDP database, comprising Gnathostomata sequences. The species were selected following the same procedure as for the Bilateria dataset, finetuning the RDP phylogeny as follows:

- Rattus and Mus are united into the monophyletic group "Rodentia", which is part of the Mammalia.
- "(3.10.6.1.1) AVES_(BIRDS)" and "(3.10.6.1.3) REPTILIA" form one subtree, called "Reptilia (incl. Aves)".
- "Reptilia (incl. Aves)" and "(3.10.6.1.2) MAMMALIA" are in one subtree, called "Amniota".
- "Amniota" and "(3.10.6.1.4) AMPHIBIA" are in one subtree, called "Tetrapoda".
- "Tetrapoda" and "(3.10.6.1.5) COELACANTHIFORME" form the group "Sarcopterygii".
- "Sarcopterygii" and "(3.10.6.1.6) ACTINOPTERYGII_(RAY-FINNED_FISHES)" comprise "(3.10.6.1) GNATHOSTOMATA_(JAWED_VERTEBRATES)".

Finally, group "(3.10.6.1) GNATHOSTOMATA_(JAWED_VERTEBRATES)" and group "(3.10.6.2) AGNATHA_(JAWLESS_VERTEBRATES)" are the relevant first two groups listed for the Chordata. We consider this tree very safe, but not as undisputed as the one presented for the Bilateria.

The group to be studied is Gnathostomata, and since this is the first group in the list, the second group ("Agnatha") on the same level (that is, "(3.10.6) CHORDATA") serves as outgroup. Species selection then follows the "take *the first two*" rule within the refined phylogeny. The same rules as in section 4.4.3 are used in case of problems. Furthermore, Trachemys scripta, the first species in the group "(3.10.6.1.3) REPTILIA", is ignored, because its phylogenetic relationship to birds on the one hand, and other "Reptilia" on the other hand, may be debated.

Considering the monophyla "Rodentia", "Reptilia (incl. Aves)", "Amniota", "Tetrapoda" and "Sarcopterygii" just described, the following table taken from the RDP Phylogeny Browser gives rise to the species found in the tree of Fig. 4.11, left.

(3.10.6) CHORDATA
(3.10.6.1) GNATHOSTOMATA_(JAWED_VERTEBRATES)
(3.10.6.1.1) AVES_(BIRDS)
....
Trdu.migrt Turdus migratorius (thrush bird, unspecified)
....
Gall.gallu Gallus gallus (chicken)
....
(3.10.6.1.2) MAMMALIA
Oryc.cunic Oryctolagus cuniculus str. New Zealand (rabbit) – *the position of this species is debated*
Oryc.cuni2 Oryctolagus cuniculus (rabbit)
Homo_sapi2 Homo sapiens (human)

Homo_sapie Homo sapiens (human)

Homo_sapi5 Homo sapiens (human)

Homo_sapi3 Homo sapiens (human)

Ratt.norw4 Rattus norvegicus (brown, common or Norway rat)

Ratt.norwe Rattus norvegicus (brown, common or Norway rat)

Ratt.norw2 Rattus norvegicus str. Sprague Dawley (brown, ... rat)

Mus_muscu4 Mus musculus (house mouse) – *fragment only*

Ratt.norw3 Rattus norvegicus (brown, common or Norway rat)

Mus_muscul Mus musculus (common or house mouse)

Mus_muscu2 Mus musculus (house mouse) – *fragment only*

(3.10.6.1.3) REPTILIA

Trch.scrpt Trachemys scripta (red-eared slider turtle; dime-store turtle)– *the position of this species is debated*

Alli.msspp Alligator mississippiensis (American alligator)

Htrd.pltyr Heterodon platyrhinos (eastern hognose snake)

Sclp.undul Sceloporus undulatus (iguanian lizard)

(3.10.6.1.4) AMPHIBIA

Xeno.laev6 Xenopus laevis (African clawed frog; South African clawed frog)

Xeno.laevi Xenopus laevis (African clawed frog; South African clawed frog)

Xeno.laev3 Xenopus laevis (African clawed frog; South African clawed frog)

Xeno.laev4 Xenopus laevis (African clawed frog)

Xeno.borea Xenopus borealis (clawed frog) – *incomplete RNA*

Xeno.laev5 Xenopus laevis (tongueless frog; African clawed frog)

Disg.picts Discoglossus pictus (painted frog)

Amby.mexic Ambystoma mexicanum (axolotl (mole salamander))

Hyla_ciner Hyla cinerea (green tree frog)

Bufo_valli Bufo valliceps (Gulf coast toad)

(3.10.6.1.5) COELACANTHIFORME

Ltmr.chlmn Latimeria chalumnae (coelacanth) – *one single species*

(3.10.6.1.6) ACTINOPTERYGII_(RAY-FINNED_FISHES)

Echr.cooke Echinorhinus cookei (prickly shark)

Squa.acant Squalus acanthias (spiny dogfish)

....

(3.10.6.2) AGNATHA_(JAWLESS_VERTEBRATES)

(3.10.6.2.1) PETROMYZONTIFORMES_(LAMPREYS)

Lptr.aepyp Lampetra aepyptera (least brook lamprey)

Ptrm.marin Petromyzon marinus (sea lamprey; marine lamprey)

(3.10.6.2.2) MYXINIFORMES_(HAGFISH)

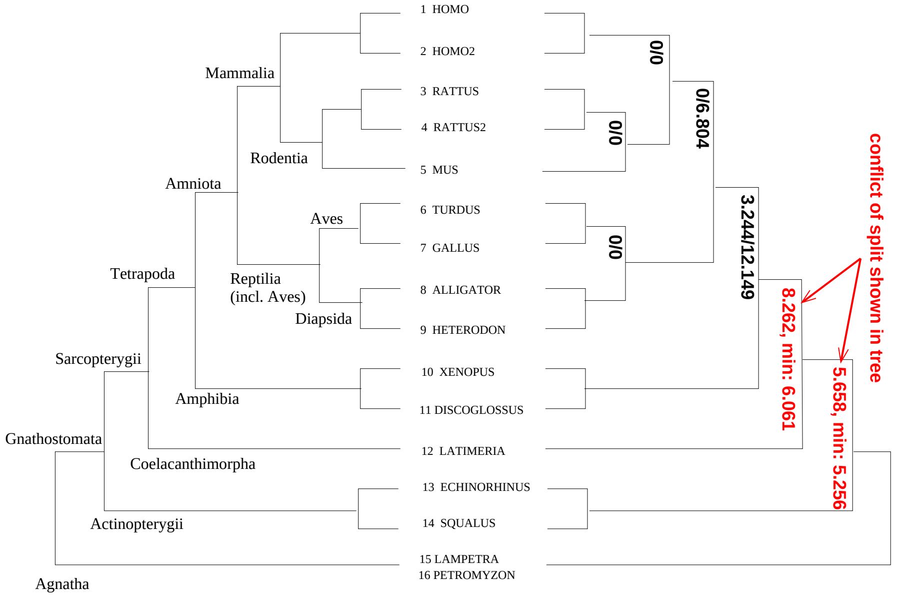
(3.10.6.3) BRANCHIOSTOMIDAE_(LANCELETS)

(3.10.6.4) UROCHORDATA_(TUNICATES)

MCOPE estimates only part of the tree shown in Fig. 4.11, right. The red labels of the first two splits indicate that these are not reconstructed correctly, even though the distance between the correct split (listed first) and the incorrect minimum-conflict one (listed next) is small: 1-12 v 13,14 has conflict 5.658, and the minimum conflict split 12 v 1-11,13,14 has conflict 5.256. Not shown are the conflict values of 12 v 1-11,13,14 (5.432) and 1-9 v 10-14 (6.746), which come next. MCOPE indicates that there is a problem, and a polytomy should be returned. In case of the next split to be determined, the correct one (1-11 v 12) has some more conflict (8.262) than the minimum-conflict one (1-9 v 10-12), with conflict 6.061. The other splits are estimated correctly, and the minimum conflict is much smaller than the conflict of any other split. Not considering columns with gap character states, we retrieve the same tree, contemplating the same set of problems.

The number of variable sites considered by MCOPE ranges from 185 for species 1-14 to 26 for species 1-5, which must be considered very low. The

Figure 4.11: "Undisputed" tree (left) and conflict tree (right) for the Gnathostomata dataset.



reason lies in the skipping of any alignment column that includes missing data.

4.4.5 Analysis of the Mammalia Dataset

The Mammalia Dataset is also taken from the RDP database, following the same procedures with two notable exceptions:

- We are interested in two debated sets of splits, the first one among the Hominids, and the second one between the Eutheria (placental mammals), Marsupialia (opossums, kangaroos, etc.) and Monotremata (platypus, echidnas).
- We select the data from the "mitochondrial DNA" part of the RDP database; there is no 18S-rDNA available for the most of the species concerned.

The following list was retrieved from the RDP database; boldface highlights the sequences selected based on our interest in the Mammalia as well as the Hominids, and the "take *the first two*" rule explained in section 4.4.3. The outgroup used is "Trachemys scripta", which is the first group listed that belongs to the "Amniota" (see section 4.4.4).

(4.4) VERTEBRATES
    (4.4.5) REPTILES
        (4.4.5.1) PARAPHYLETIC_REPTILE_GROUP_I – *uncertain phylogeny*

(4.4.5.2) TURTLES

(4.4.5.2.1) SLIDER_TURTLE

Trch.scr_M Trachemys scripta (red-eared slider turtle; slider turtle; dime-store turtle) – mitochondrion

....

....

(4.4.6) MAMMALS

(4.4.6.1) MARSUPIALS

(4.4.6.1.1) MARSUPIAL_SUBGROUP_I

Phln.ori_M Phalanger orientalis (gray common cuscus) – mitochondrion

Phln.car_M Phalanger carmelitae (mountain cuscus) – mitochondrion – *fragment only*

Trcs.vul_M Trichosurus vulpecula (brush-tailed possum) – mitochondrion – *fragment only*

Phcl.cin_M Phascolarctos cinereus (koala) – mitochondrion – *fragment only*

Phcl.ci2_M Phascolarctos cinereus (koala) – mitochondrion

....

(4.4.6.1.2) MARSUPIAL_SUBGROUP_II – *uncertain phylogeny*

(4.4.6.1.3) MARSUPIAL_SUBGROUP_III – *uncertain phylogeny*

(4.4.6.1.4) MARSUPIAL_SUBGROUP_IV – *uncertain phylogeny*

(4.4.6.2) MONOTREMATES_(EGG-LAYING_MAMMALS)

(4.4.6.2.1) PLATYPUS

Orrh.an3_M Ornithorhynchus anatinus (duckbill platypus) – mitochondrion

Orrh.an4_M Ornithorhynchus anatinus (duckbill platypus) – mitochondrion

Orrh.an2_M Ornithorhynchus anatinus (duckbill platypus) – mitochondrion – *fragment only*

Orrh.ana_M Ornithorhynchus anatinus (duckbill platypus) – mitochondrion – *fragment only*

(4.4.6.2.2) ECHIDNAS – *only fragments found*

(4.4.6.3) RODENT_GROUP_I

(4.4.6.3.1) RODENT_SUBGROUP_I

Prmy.pol_M Peromyscus polionotus (deer mouse) – mitochondrion

Prmy.kee_M Peromyscus keeni (deer mouse or North American wood mouse) – mitochondrion

....

(4.4.6.3.2) RODENT_SUBGROUP_II

Onch.are_M Onychomys arenicola (Chihuahuan grasshopper mouse) – mitochondrion

Onch.leu_M Onychomys leucogaster (northern grasshopper mouse) – mitochondrion

....

....

....

(4.4.6.19) PRIMATES

(4.4.6.19.1) HOMINIDS

Pan_tro5_M Pan troglodytes (common chimpanzee) – mitochondrion

Pan_tro4_M Pan troglodytes (common chimpanzee) – mitochondrion

Pan_trog_M Pan troglodytes (common chimpanzee) – mitochondrion – *fragment only*

Pan_tro3_M Pan troglodytes (central African common chimpanzee) – mitochondrion

Homo_sa9_M Homo sapiens (human) – mitochondrion

Pan_tro2_M Pan troglodytes (common chimpanzee) – mitochondrion

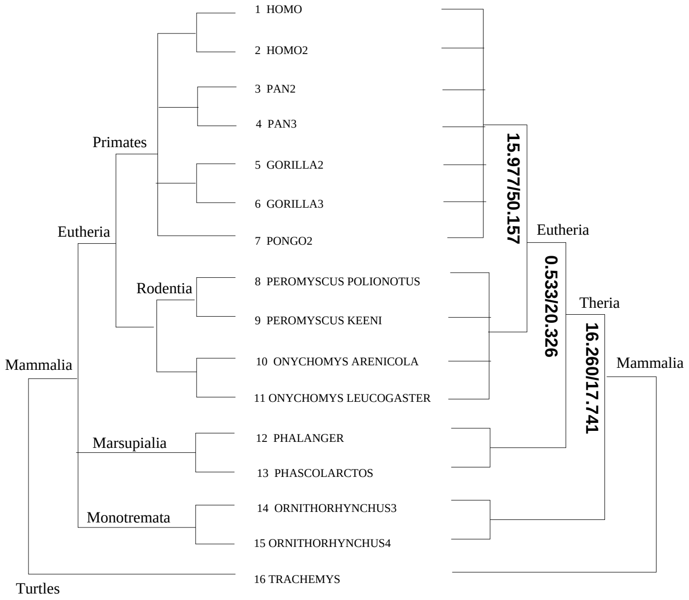
Pan_pani_M Pan paniscus (pygmy chimpanzee) – mitochondrion – *fragment only*

Pan_pan2_M Pan paniscus (pygmy chimpanzee) – mitochondrion – *unreadable genbank file*

Gorl.go2_M Gorilla gorilla (gorilla) – mitochondrion

Gorl.gor_M Gorilla gorilla (lowland gorilla) – mitochondrion – *fragment only*
Gorl.go3_M Gorilla gorilla (Western lowland gorilla) – mitochondrion
Homo_sa3_M Homo sapiens (human) – mitochondrion
Homo_sap_M Homo sapiens (human) – mitochondrion
Homo_sa2_M Homo sapiens (human) – mitochondrion
Homo_sa4_M Homo sapiens (human) – mitochondrion
Homo_sa8_M Homo sapiens (human) – mitochondrion
Homo_sa5_M Homo sapiens (human) – mitochondrion
Homo_sa7_M Homo sapiens (human) – mitochondrion
Pong.py2_M Pongo pygmaeus (orangutan) – mitochondrion
(4.4.6.19.2) NON-HOMINID_PRIMATES – *only fragments found*

Figure 4.12: Unresolved tree (left) and minimum conflict tree (right) for the Mammalia dataset (mitochondrial DNA).

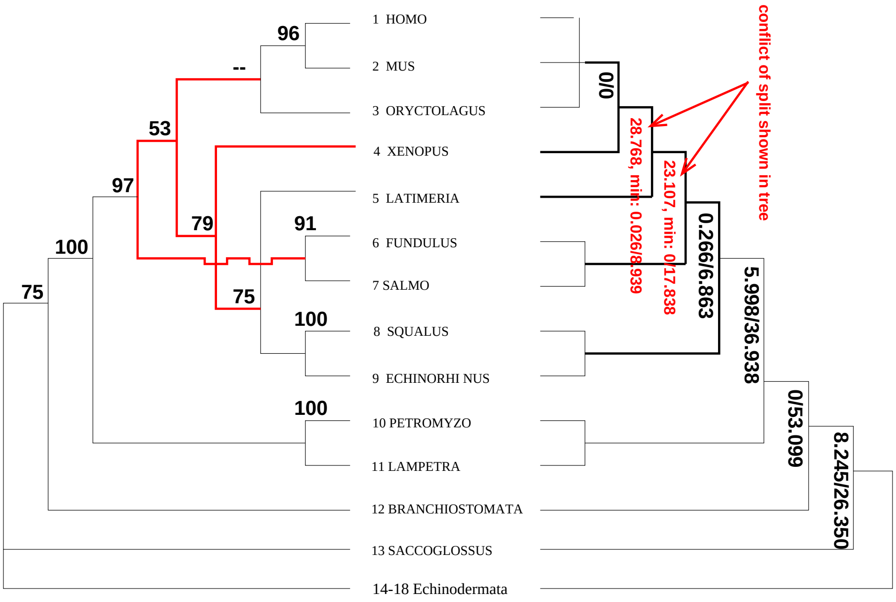


The result of the selection process are the species included in the tree of Fig. 4.12, left; the tree shown may be debated as far as the position of the Monotremata and the various hominids is concerned, and we use polytomies to indicate this. On the right, the minimum conflict tree is displayed; the split 1-13 v 14,15 (Theria versus Monotremata) has minimum conflict, although it is closely followed by 12 v 1-11,13-15 (conflict 17.741), 1-12 v 13-15 (conflict 20.417), 12,13 v 1-11,14,15 (conflict 22.280), 8-11 v 1-7,12-15 (conflict 25.960) and 1-11 v 12-15 (conflict 28.732). The last split listed refers to the "Marsupionta" hypothesis, proposing a monophyletic group of Marsupialia and Monotremata [15, 16, 25]. The other splits (Eutheria versus Marsupialia, and Primates versus Rodentia) are well supported. No further resolution is possible for the primates; there are seven splits with zero conflict. This is no surprise: While 413 variable sites are given in the alignment of 1-15, and 294 are still available in the alignment of 1-11, no resolution is possible for the 107 variable columns in the projected alignment of the primate species 1-7.

4.4.6 Analysis of the Chordata Dataset

The Chordata dataset is a subset of the Metazoa dataset analyzed by Rödding and Wägele [26]. A projection of a PAUP version 3.1 bootstrap run using parsimony [35] of all 98 species is given in Fig. 4.13, left. The choice of species is close to the one of the Gnathostomata dataset, but the underlying alignment is provided by the authors, who used a computerized procedure (ClustalW, [37]) and then maximized the number of invariable sites by eye. We note that in addition to the groups already discussed, species 1-13, species 1-12 and species 1-7 in this dataset are usually considered to be monophyletic.

Figure 4.13: Doubtful tree (left) and conflict tree (right) for the Chordata dataset.



The parameters of the PAUP run have been:

- 500 bootstraps
- Heuristic Tree Search
- Nearest Neighbor Interchange
- Addition sequence simple, reference species Mus Musculus.

The run assigns bootstrap confidence of 53% to the presumably incorrect split 1-5,8,9 v 6,7 (see Fig. 4.13, left, including bootstrap support values), and we assume that leftover character states due to more rapid evolution of Fundulus and Salmo trigger the incorrect tree topology, since MCOPE gets the tree right up to this point (Fig. 4.13, right). Thereafter, the conflict tree is as incorrect as the PAUP-tree, favoring a zero-conflict split 1-3 v 4-7 over the correct split 1-5 v 6,7, which obtains a conflict of 23.107. Without considering gaps, calculations run very similar, except that split 1-11 v 12 is no longer minimum, giving way to 10,11 v 1-9,12, which is not a plausible bipartition into monophyletic subtrees. For the Chordata dataset, MCOPE recognizes 662 variable sites in the beginning, and this number goes down to 341 for species 1-9, and to 323 for the subtree of species 1-7. In the end, 50 variable columns are left in the alignment of the sequences of species 1-3.

4.5 MCOPE Advantages and Disadvantages

Our method, "minimum conflict phylogeny estimation" (MCOPE), has the following advantages, which have been outlined in section 1.2, unless noted otherwise.

- MCOPE avoids short-branch attraction (erosion).
- MCOPE is fast due to its divide-and-conquer approach; many species can be handled simultaneously, and no search through the space of tree topologies is necessary.
- MCOPE returns rooted trees, which are more informative than unrooted trees.
- MCOPE is simple and transparent.
- MCOPE needs no modelling of substitution rates, etc.
- MCOPE is rather robust to species sampling problems, see section 3.9.
- MCOPE returns results that do not depend in any way on the input order of the sequences.

The following disadvantages should be noted:

- MCOPE does not deal with protein sequences yet.
- MCOPE, as currently implemented, is slower than necessary because it is all written in a scripting language (Perl), without using the numerical data handling package that Perl now offers.
- MCOPE may fall victim to long branch attraction due to the accumulation of convergences, see section 3.5.2.
- MCOPE may fall victim to parallel erosion and speciation, see section 3.5.2.

4.6 Future Work

For updates on our work, and more applications of our method to biological and artificial datasets, please watch the URL
`http://bibiserv.techfak.uni-bielefeld.de/mcope/`.

The following is a list of possible improvements.

- Method and algorithm
  - There needs to be an assessment on how far the split of lowest conflict is away from the follow-on ones, and how significant this distance is.
  - If this distance is insignificant, the method may flag reticulate (net-like) evolution, and/or it may follow up on more than one low-conflict split, analyzing the subtrees in parallel, and possibly combining the results of the analyses.
  - The detection of convergence accumulation may be possible by using the detection of erosion in the complementary set of species as a flag.
  - We would like to estimate the acceptable validity estimate from the dataset.
  - Theoretical results may be very useful, following up on the very general model introduced in section 2.3. Philosophical issues can then be tackled more directly. MCOPE uses the local similarity of character states in an intelligent way, just like parsimony. Therefore, we claim that it is not a phenetic method in any reasonable sense of the word, because is it does not just look at the (overall) similarities between character states. Theoretical results could add some deeper understanding of these issues.
  - Simulation studies may include a direct comparison with standard phylogeny estimation methods.
  - It may be possible to avoid the heuristic search of split space, and do a simultaneous analysis of all the patterns in a spectrum. It may be possible to analyze these for erosion, independently of any split.
  - The handling of gap character states (in particular terminal gaps) and unknown nucleotides / missing data should be improved. In particular, the failure to distinguish terminal from nonterminal (interior) gaps may be responsible for some of the errors reported for real datasets whenever alignment columns with gap character states are not ignored.
  - The method should be generalized to protein data.
- Implementation issues
  - A better automated output of trees in various formats is needed.

- Since the prototyping phase is now over, memory consumption and speed should be cut by several orders of of magnitude, investing just a few days of finetuning work.
- The crude databases used for maintaining parameter information and associated results need to be improved.
- A standalone distribution needs to be developed, most likely based on a standalone server/browser setup.

4.7 A Short Description of the MCOPE Software

MCOPE software is written using the Perl programming language [43]. It consists of object-oriented modules for alignment manipulation (see [3]), phylogeny manipulation, phylogeny exploration and corresponding alignment visualization. The phylogeny explorer allows the user to specify which splits of a given phylogeny s/he wishes to inspect (sorted and filtered by weak weight and conflict, including an optional search for better splits via conflict-based species exchange). The explorer also manages the recursion, preceded by another optional filtering step, and allows for each filtering step the setting of a minimum and maximum number of splits to be retained. For any path, the explorer can return results as hypertext and as graphics; in both cases cumulative reports can be returned, effectively listing conflict spectra as linked tables or histograms.

Chapter 5

Acknowledgement

It is not easy to write acknowledgements, and finding the right tone, especially not at 2 am on the day of submission. Let's hope the best... Let's hope I don't miss anyone, and that I don't say anything stupid... (And let's hope this gets printed in time... :-)

I am grateful for all the support received, from very many people.

Prof. Robert Giegerich has been an excellent advisor, giving me the opportunity to work in a research group with great minds, an excellent atmosphere & best working conditions. I have learned a lot from him, on very many scientific topics. Prof. Wolfgang Wägele introduced me to the world of phylogeny estimation, and he has done a wonderful job as the co-supervisor of this thesis. I'm looking forward to continue working with both – that is a ”best choice”.

I'm thankful to Prof. Andreas Dress for his support, especially in the early days of my Bielefeld experience, and to Prof. Gerhard Sagerer, for chairing the committee and permitting color printouts of the thesis on the printer in his research group. (A big thank-you regarding color printouts also goes to Prof. Helge Ritter.) Moreover, I'd like to thank Prof. Friedrich Götze for some valuable advice.

I'm happy that I could interact with many very good people in Bielefeld. In particular, these are Christian Büschking, Dirk Evers, Dr. Stefan Kurtz, Dieter Lorenz, Dr. Enno Ohlebusch, Chris Schleiermacher, Alexander Sczyrba and Dirk Strothmann. Dirk Evers and Alexander Sczyrba have given valuable feedback on draft versions of the thesis. I've also got very good feedback on the text from Ingo Busse, Bielefeld, and Dr. Christoph Held, who is currently moving to Bochum. In Prof. Wolfgang Wägele's lab in Bochum, I've also got a lot of support from Dr. Hermann Dreyer and Dr. Heike Wägele.

I would like to thank the Perl community for the open software used for this work; in particular I'd like to thank Tim Pearson for the PGPLOT plotting library, Karl Glazebrook for the PGPERL interface to PGPLOT, Steffen Beyer for his Perl bitvector implementation, and M. Constant and P. Gordon for some programming hints.

This work was funded in part by a stipend from the ”Deutsche Forschungsgemeinschaft – Graduiertenkolleg Strukturbildungsprozesse”. Last but most importantly, I'm grateful to Susanne Dohmann, who has been the patient one in recent months, and who will hopefully see me a lot more in years to come. This work is dedicated to my parents, based on necessity and reverence.

Index

Bibliography

[1] D.J. Aldous. *Stochastic Models and Descriptive Statistics for Phylogenetic Trees, from Yule to Today.* Preprint available at http://www.stat.berkeley.edu/users/aldous/bibliog.html, 2000.

[2] J. A. Cavender and J. Felsenstein. Invariants of phylogenies in a simple case with discrete states. *Journal of Classification*, 4:57–71, 1987.

[3] S.A. Chervitz, G. Fuellen, C. Dagdigian, S. E. Brenner, Birney E., and I. Korf. Bioperl: Standard Perl modules for bioinformatics. *BITS Journal*, 1, 1999. Article URL: http://www.bitsjournal.com/bioperl.html, Journal URL: http://www.bitsjournal.com/.

[4] J. Darnell, H. Lodish, and D. Baltimore. *Molecular Cell Biology.* W.H. Freeman & Co., New York, 1995.

[5] T. Dobzhansky. Nothing in biology makes sense except in the light of evolution. *American Biology Teacher*, 35:125–129, 1973.

[6] M. Friedrich and D. Tautz. Ribosomal DNA phylogeny of the major extant arthropod classes and the evolution of myriapods. *Nature*, 376:165–167, 1995.

[7] G. Fuellen. Multiple alignment. *Complexity International*, 4, 1997. Article URL: http://www.csu.edu.au/ci/vol4/mulali/mulali.html, Journal URL: http://www.csu.edu.au/ci/.

[8] G. Fuellen and J.W. Wägele. Phylogeny inference by minimum conflict. In W. Gaul and R. Decker, editors, *Classification and Information Processing at the Turn of the Millenium, Proceedings of the 23rd Annual Conference of the Gesellschaft für Klassifikation, March 10-12, 1999, University of Bielefeld*, pages 377–385, 2000.

[9] M.D. Hendy and D. Penny. Spectral analysis of phylogenetic data. *J. Classification*, 10:5–24, 1993.

[10] W. Hennig. *Phylogenetic Systematics.* University of Illinous Press, 1966.

[11] D. M. Hillis and J. P. Huelsenbeck. Support for dental HIV transmission. *Nature*, 369:24–25, 1994.

[12] E. C. Holmes, P. L. Bollyky, S. Nee, A. Rambaut, G. P. Garnett, and P. H. Harvey. Using phylogenetic trees to reconstruct the history of infectious disease epidemics. In P. H. Harvey, A. J. Leigh Brown, J. Maynard Smith, and S. Nee, editors, *New Uses for New Phylogenies*, pages 169–186. Oxford University Press, New York, 1996.

[13] D. Huson, S. Nettles, L. Parida, T. Warnow, and S. Yooseph. The disk-covering method for tree reconstruction. In R. Battiti and A.A. Bertossi, editors, *Proceedings of "Algorithms and Experiments" (ALEX), Trento, Italy*, pages 62–75. 1998.

[14] D.H. Huson, S.M. Nettles, and T.J. Warnow. Disk-covering, a fast-converging method for phylogenetic tree reconstruction. *J. Computational Biology*, 6, 1999.

[15] A. Janke, N.J. Gemmell, G. Feldmaier-Fuchs, A. von Haeseler, and S. Paabo. The mitochondrial genome of a monotreme - the platypus (ornithorhynchus anatinus). *J. Mol. Ecol*, 42:153–159, 1996.

[16] A. Janke, X. Xu, and U. Arnason. The complete mitochondrial genome of the wallaroo (macropus robustus) and the phylogenetic relationship among monotremata, marsupialia, and eutheria. *Proc. Natl. Acad. Sci. USA*, 94:1276–1281, 1997.

[17] J. A. Lake. A rate-independent technique for analysis of nucleic acid sequences: evolutionary parsimony. *Molecular Biology and Evolution*, 4:167–191, 1987.

[18] S. Ludwig, C. Scholtissek, and W. M. Fitch. Analysis of influenza virus nucleoproteins for the assessment of molecular genetic mechanisms leading to new phylogenetic virus lineages. *Archives of Virology*, 131:237, 1993.

[19] B.L. Maidak, J.R. Cole, T.G. Lilburn, Parker C.T., Saxman P.R., Stredwick J.M., Garrity G.M., Li B., Olsen G.J., Pramanik S., Schmidt T., and Tiedje J. The RDP (Ribosomal Database Project) continues. *Nucleic Acids Res.*, 28:173–174, 2000.

[20] A. Meyer. The evolution of body plans: Hom/hox cluster evolution, model systems, and the importance of phylogeny. In P. H. Harvey, A. J. Leigh Brown, J. Maynard Smith, and S. Nee, editors, *New Uses for New Phylogenies*, pages 203–216. Oxford University Press, New York, 1996.

[21] C. Moritz. Uses of molecular phylogenies for conservation. In P. H. Harvey, A. J. Leigh Brown, J. Maynard Smith, and S. Nee, editors, *New Uses for New Phylogenies*, pages 203–216. Oxford University Press, New York, 1996.

[22] S. P. Otto, M. P. Cummings, and J. Wakeley. Inferring phylogenies from DNA sequence data: the effects of sampling. In P. H. Harvey, A. J. Leigh Brown, J. Maynard Smith, and S. Nee, editors, *New Uses for New Phylogenies*, pages 103–115. Oxford University Press, New York, 1996.

[23] C.Y. Ou, Ciesielski C.A., Myers G., Bandea C.I., Luo C.C., Korber B.T., Mullins J.I., Schochetman G., Berkelman R.L., Economou A.N., and et al. Molecular epidemiology of HIV transmission in a dental practice. *Science*, 256:1165–71, 1992.

[24] C.Y. Ou, Takebe Y., Weniger B.G., Luo C.C., Kalish M.L., Auwanit W., and et al. Independent introduction of two major HIV-1 genotypes into distinct high-risk populations in thailand. *Lancet*, 341:1171–4, 1993.

[25] D. Penny and M. Hasegawa. The platypus put in its place. *Nature*, 387:549–550, 1997.

[26] F. Rödding and J.W. Wägele. Origin and phylogeny of the metazoans as reconstructed with rDNA sequences. *Progr. Mol. Subcell. Biol.*, 21, 1998.

[27] B. Rost and C. Sander. Prediction of protein structure at better than 70% accuracy. *J. Mol. Biol.*, 232:584–599, 1993.

[28] N. Saitou and M. Nei. The neighbor-joining method: A new method for reconstructing phylogenetic trees. *Mol. Biol. Evol.*, 4(4):406–425, 1987.

[29] B. Salisbury. Strongest evidence: maximum apparent phylogenetic signal as a new cladistic optimality criterion. *Cladistics*, 15:137–149, 1999.

[30] P. M. Sharp, D. L. Robertson, and B. Hahn. Cross-species transmission and recombination of 'Aids' viruses. In P. H. Harvey, A. J. Leigh Brown, J. Maynard Smith, and S. Nee, editors, *New Uses for New Phylogenies*, pages 134–152. Oxford University Press, New York, 1996.

[31] T. Spears, L.G. Abele, and M.A. Applegate. Phylogenetic studies of cirripedes and selected relatives (thecostraca) based on 18S rDNA sequence analysis. *J. Crust. Biol.*, 14:641–656, 1994.

[32] J. Stoye, D. Evers, and F. Meyer. Rose: generating sequence families. *Bioinformatics (formerly CABIOS)*, 14:157–163, 1998.

[33] K. Strimmer and A. von Haeseler. Quartet puzzling: a quartet maximum likelihood method for reconstructing tree topologies. *Mol. Biol. Evol.*, 13:964–969, 1996.

[34] T. Swinscow. *Statistics at Square One*. BMJ Publishing Group, 1997. URL: http://www.bmj.com/collections/statsbk/index.shtml; Revised by M J Campbell.

[35] D.L. Swofford. Paup: Phylogenetic analysis using parsimony, version 3.1. Illinois Nat. Hist. Survey, Champaign, 1993.

[36] D.L. Swofford, G.J. Olsen, P.J. Waddell, and D.M. Hillis. Phylogenetic infererence. In D.M. Hillis, C. Moritz, and B.K. Mable, editors, *Molecular Systematics*, pages 407–514. Sinauer Associates Inc., Sunderland, MA, USA, 1996.

[37] J.D. Thompson, D.G. Higgins, and T.J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, positions-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680, 1994.

[38] C. Tuffley and M. Steel. Links between maximum likelihood and maximum parsimony under a simple model of site substitution. *Bull. Math. Biol.*, 59:581–607, 1997.

[39] Y. Van de Peer, S. Chapelle, and R. De Wachter. A quantitative map of nucleotide substitution rates in bacterial rRNA. *Nucleic Acids Research*, 24:3381–91, 1996.

[40] Y. Van de Peer, J. Jansen, P. De Rijk, and R. De Wachter. Database on the structure of small ribosomal subunit RNA. *Nucleic Acids Research*, 25:111–6, 1997.

[41] J.-W. Wägele. First principles of phylogenetic systematics, a basis for numerical methods used for morphological and molecular characters. *Vie Milieu*, 46(2):125–138, 1996.

[42] J.W. Wägele. Identification of apomorphies and the role of groundpatterns in molecular systematics. *J. Zoo. Syst. Evol. Research*, 34:31–39, 1996.

[43] L. Wall, T. Christiansen, and L. Schwartz. *Programming Perl.* O'Reilly, 1996.

[44] T. Warnow and J. Kim. *Computational and statistical challenges involved in reconstructing evolutionary trees.* ISMB tutorial No. 4., 1999.