

5 Computational methods and tools

Introductory remarks by the chapter editor, Joris van Zundert

This chapter may well be the hardest in the book for those that are not all that computationally, mathematically, or especially graph-theoretically inclined. Textual scholars often take to text almost naturally but have a harder time grasping, let alone liking, mathematics. A scholar of history or texts may well go through decades of a career without encountering any maths beyond the basic schooling in arithmetic, algebra, and probability calculation that comes with general education. But, as digital techniques and computational methods progressed and developed, it transpired that this field of maths and digital computation had some bearing on textual scholarship too. Armin Hoenen, in section 5.1, introduces us to the early history of computational stemmatology, depicting its early beginnings in the 1950s and pointing out some even earlier roots. The strong influence of phylogenetics and bioinformatics in the 1990s is recounted, and their most important concepts are introduced. At the same time, Hoenen warns us of the potential misunderstandings that may arise from the influx of these new methods into stemmatology. The historical overview ends with current and new developments, among them the creation of artificial traditions for validation purposes, which is actually a venture with surprisingly old roots.

Hoenen's history shows how a branch of computational stemmatics was added to the field of textual scholarship. Basically, both textual and phylogenetic theory showed that computation could be applied to the problems of genealogy of both textual traditions and biological evolution. The calculations involved, however, were tedious, error-prone, hard, and cumbersome. Thus, computational stemmatics would have remained a valid but irksome way of dealing with textual traditions if computers had not been invented. Computers solve the often millions of calculations needed to compute a hypothesis for a stemma without complaint. They do so with ferocious speed and daunting precision. But it remains useful to appreciate that this is indeed all they do: calculate. The computer – or algorithm – does not have any grasp of the concepts or problems that it is working on. Nowhere in the process leading from variant data to a stemmatic hypothesis does any software or hardware realise that it is working on a textual tradition or genetic material. It has no feelings about that work and – more saliently – is indifferent to the quality, correctness, or meaning of the result it calculates. It is especially for this last reason that textual scholars should take note of the methods and techniques involved in calculating stemmata, even if the maths may not always be palatable work. Computer code and chips process data and yield some result or other. None of the nouns in the previous sentence somehow becomes inherently neutral, objective, and correct by virtue of being digital or mathematical in nature. If an algorithm contains a calculation error, the computer will repeat that error faithfully a billion times at lightning speed. Thus, it follows that we can only trust digital tools and computational methods if we can trust their theoretical and mathematical underpinnings, if

we can trust how they are implemented as software code, and if we understand how we are translating the computational concepts back into philological ones.

Providing a basic insight into the concepts, theory, and mathematical underpinnings of graph theory and computational tree construction is the aim of sections 5.2 and 5.3. These sections, where Staedler, Manafzadeh, and Roos try to explain the various methods and techniques that exist for the computational creation of stemmata, may make for challenging reading. Many new concepts are introduced – reticulation events, character-state matrices, and tree scores, to name but a few – some more relevant to stemmatology than others, but all important when it comes to understanding how the techniques of phylogenetics operate and what their relevance to stemmatology is. Aware of the dangers of overstressing the direct application of computational techniques from bioinformatics in the domain of stemmatology, Roos points out that the methods under discussion do not, in fact, produce stemmata; rather, they produce graphs, trees, or networks that can be regarded as hypotheses for stemmata. He then details the methods most commonly used to generate these stemma hypotheses. This handbook confines itself to providing a basic understanding of the computational methods that are involved with tree building and visualisation. Explaining all the fine-grained fundamental mathematical intricacies of parsimony tree scores, maximum likelihood, UPGMA, neighbour-joining, and so forth is beyond what will fit on its pages, and for this we refer to additional reading. The sections here aim to provide some intimations and a very basic but very necessary understanding about the calculations that are employed when inferring graphs and trees – accompanied by some warnings about the limitations of these methods.

Over the course of time, philologists, software engineers, and computer scientists have ventured to create tools that embrace the mathematical principles of phylogenetics and stemmatics to provide ready-to-use software tools for evolutionary biologists and textual scholars. These tools provide (somewhat) easier access to the calculations needed to create stemma hypotheses. Digital tool development unfortunately has its own problems of life-cycle management, sustainability, and compatibility. What worked five years ago may fail on the newest operating systems, tools sometimes get abandoned for economic and institutional policy reasons that have nothing to do with their actual usefulness or capabilities, and in general the digital landscape changes frighteningly quickly. Any author who has ever added a chapter or section on digital tools to a textbook knows that these efforts are at risk of becoming obsolete in part or in whole even when the copies are running off the press. Nevertheless, a chapter on computational methods and tools cannot be complete without such a list, even if it may become out of date with its very inception. Hønen faithfully assembles a list of the tools (5.4) most visibly in use in the field of computational stemmatics (and some related work) today. Some tools, such as *Phyloip* or *PAUP**, have been around for years and may still be around for ages to come. Others may linger and die off. Some have been born very recently and still need to

prove whether they will make it through their toddler years. All this aside, however, the list provides an impressively extensive overview of the tools that currently figure in the centre of computational stemmatics.

Finally, Jean-Baptiste Guillaumin directs our attention to the criticisms that computational approaches to stemmatology have met over time. The final section of this chapter (5.5) treats a number of well-known problems that are real problems from the perspective of both philologists (e.g. “How well does all this computation match the genealogical process of text descent?”) and computationalists (e.g. “How do we model witnesses as internal nodes?”). Computational stemmatology is a young field, and its methods are still – and should be – in development, in flux, and under criticism. Some criticism pertains to basic concepts such as what we mean exactly by the distance between two (or more) texts: how such a distance is computed and what an appropriate measure for it could be. However, more advanced parts of the computational approach also meet with criticism. Computational methods to date tend to take all data as equally valuable. But philologists realise that some readings should be weighted more heavily than others; and, vice versa, what computational methods may regard as statistical noise may very well be pivotal readings that reveal genealogy to a philologist. And what about the prevalence of bifurcation in all computational approaches? As Guillaumin shows, both philologists and computationalists are taking these matters increasingly more seriously. The computational approach to stemmatology appears not to cut corners to a quick and dirty win. Rather, the particulars of textual genealogy prove a challenge to computer scientists that both computer scientists and philologists are engaging with deeply, and with mutual respect for the expertise on both sides.

5.1 History of computer-assisted stemmatology

Armin Hoenen

This section covers the history of the field, beginning roughly at a time when the first computers became commercially available and extending up to the present. Throughout this history, due to technological and epistemological developments, the umbrella term for the subject of this chapter has seen terminological variation: besides “computational stemmatology”, a slightly more appropriate term is “stemma-tology aided or assisted by computers”. The computer cannot conduct certain tasks, such as digitising a manuscript text, on its own, and humans have to supervise the process of arriving at a stemma. “Digital stemmatology” is another synonym in use.

5.1.1 General remarks, early history, and consolidation (1950s–1990s)

There are many ways in which computers can be used for stemmatological purposes; a strict interpretation of the term implies the application of software where

the input is a collection of digitised manuscript texts (or similar data) and the outcome a stemma. In a wider sense, the term may refer to all methods, processes, and approaches using the computer for any task connected with solving the question of how to reconstruct or display the history of a number of extant textual items related through processes of copying.

Shortly after the onset of publicly available computation, as early as 1957, John W. Ellison wrote a thesis at Harvard entitled “The Use of Electronic Computers in the Study of the Greek New Testament Text” (1957). He used the computer to group manuscripts and compare those groupings to established ones. Other early works, especially in French and English, theoretically elaborating calculations and ideas for algorithms and applying computer programs were, for instance, Griffith (1968) and Froger (1968). Glenisson et al. (1979) published the proceedings of a conference on the application of computers to the field of textual criticism. The application of the computer in those days seems to have been focused especially on variant and manuscript groupings and tables, although holistic approaches going as far as producing stemmata were already present too.

Poole (1974) wrote a program in Algol60 intended for stemma production, albeit without visual output of a stemma but with tabulation according to witness age instead. It was based on previous research (notably Froger 1968 and Griffith 1968) and tested on a real sample. Here, *automation* was presented as a holistic endeavour that included an algorithmic decision as to what variant was contained in what lost witness. Most publications up to this time were closely linked to manual philological practice (mathematically formalised to some extent), although not restricted to the Lachmannian genealogical method in terms of theory. For instance, the method of Quentin (1926) looks at all triples of extant manuscripts and determines their relationships (see 2.3.5). It was the basis for an implementation by Zarri (1976) which, for the first time, produced an unrooted tree. With the steady growth of processing capacity, in addition to a growing number of publications on automated stemma production, more subtasks became subject to experimental support by computers. Haigh (1970), for instance, invented a rooting algorithm. Improved collation and alignment algorithms were developed in computer science and bioinformatics. Robinson (1994) presented a program called Collate which supported philologically adapted, semi-automated collation. A descendant of Collate, CollateX (see Haentjens Dekker et al. 2015), now finally collates all by itself. The trend of more and more subtasks being delegated to the computer continues. Den Hollander (2004), for instance, demonstrated a successful method for detecting exemplar shift (see 4.4.6.1). Computers were employed to simulate the development in time of stemmata beginning in the 1980s, either by building up artificial genealogies simulating the concurrent texts or variant configurations (Flight 1992, 1994) or by simulating their growth abstractly (Weitzman 1982, 1987; Hoenen 2016).

5.1.2 Phylogenetic methods (1990s and beyond)

Because of the enormous influence and vast application of phylogenetic methods (see 8.1) in the preceding decades, sections 5.1.2.1–3 will present the historical transfer of methodology from bioinformatics to stemmatology, criticism which has arisen, and an outlook on the future.

5.1.2.1 Phylogenetic methods: Inner developments and parallels with stemmatology

Before describing the transfer of methods to stemmatology, we will briefly outline some developments within bioinformatics from the time when bioinformatics and stemmatology coexisted but had not been related much to each other yet. Prior to molecular methods (e.g. DNA sequencing), the dominant paradigm in phylogenetics was that of cladistics (see Hennig 1966). When applying cladistics, biologists ideally chose phenotypic traits of the group of species under scrutiny. For example, the diameter of nostrils can be classed in several groups, such as a “group 0” (up to 3 cm), “group 1” (3–5 cm), and “group 2” (larger than 5 cm). Additionally, the form of the cranium can be categorised binarily as “0” (rounded) or “1” (angular). Each species would then be characterised by an array of character states. In our fictitious two-character example, such sequences could be “00”, “01”, “10”, “11”, “20”, or “21”. These categories and choices, when devised and made reasonably, are in principle similar to the identification of significant errors common in philology. Indeed, both disciplines agreed on many things, for instance on using only shared innovations as a basis for classification. Nonetheless, there was little methodological exchange at this time. This “old” cladistics paradigm may have been more comparable to the methods used in stemmatology just before the bioinformatic influx.

Then, the paradigm in biology changed radically with cheap DNA sequencing. Cladistic choice-based methods were abandoned almost completely because molecular methods are less subjective and had become less work-intensive. Since DNA makes the underlying genotype explicit and because of the sheer amount of data in a DNA strand, the new approach is more informative and depends less on arbitrary (i.e. subjective) categorisation.

Unfortunately, we cannot assume that the superior effectiveness of molecular methods in biology – as compared to the earlier cladistics – carries over into stemmatology. The now well-established bioinformatic methods outperformed conventional cladistics because of the newly available molecular data. In other words, the superiority of molecular methods is primarily due to the amount and precision of the input data. The radical paradigm shift superseding cladistic methods was brought about in biology by a new source of input data: DNA, making the older character-data-based classification (choosing certain traits) obsolete. This new type of input data is not present in stemmatology. Of course, we do not suddenly get

more or different data by assuming that a transformation of collations is a kind of pseudo-DNA.

If insufficiently recognised, these differences can lead to the risk of methodological misunderstanding. Biologists cannot expect stemmatologists to adopt their current methodologies as inherently better than cladistics or assume that the new methods will self-evidently supersede the “old” paradigms (which to stemmatology are not “old”, since here no input-data revolution like the one replacing chosen character data by sequenced DNA has taken place, and it is unlikely that something similar will ever happen in stemmatology). Moreover, stemmatologists should, of course, not readily assume that the new methods are more objective or more adequate solutions for their problems of text genesis just because they work better in another field. Understanding the validity and quality of such new methods for another problem domain requires thorough testing and evaluation, and very possibly adaptation to the techniques involved. But, once these techniques have been mastered and are no longer merely “black boxes”, the advantage may lie in the quick availability of results from phylogenetic programs, making it possible to test and contrast more stemmatic hypotheses.

To sum up: the application of phylogenetic methods should be exploited for text-critical purposes as they add the possibility of quickly producing many alternative hypotheses, but their biological provenance and prerequisites must be understood well and reflected on to avoid the dangers of a possibly misapplied model.

5.1.2.2 Transfer to stemmatology

In the early 1990s, the field of stemmatology began to change due to the introduction of phylogenetic methods from evolutionary biology, although similarities between the disciplines of biology and philology had been outlined earlier, for instance by Cameron (1987) on a theoretical level. Lee (1989) first introduced phylogenetic methods in practice to the field, using the software package Phylip (Felsenstein 1989) to generate a stemma automatically. In the years that followed, many methodologically similar publications ensued and eventually came to dominate the field in the early 2000s. Presumably the most famous contribution appeared in *Nature*: Barbrook et al. (1998). The most-often applied software packages were PAUP/PAUP* (Swofford 1998) and SplitsTree (Huson 1998); the most widely applied algorithms were parsimony, split decomposition, and neighbour-joining. Techniques such as bootstrapping and generating consensus trees were widely used. A comparable development took place in historical linguistics (see 8.2). Other useful insights or models stemming from analogies with biology have been discussed in and introduced to computational stemmatology – for instance the molecular clock, or error distribution patterns and models of errors (Spencer and Howe 2001) that exhibit similarities with mutation rates and places (see also Windram, Spencer, and Howe 2006). Investigations with a closer link to traditional methods used in textual criticism continued to exist but became rarer after the 1990s (cf. e.g. Salemans 2000).

Finally, the use of neighbour-nets is something that can be viewed as a genuine bioinformatic innovation transferred to stemmatology, and as such as an addition that complements the existing visualisation and analytical arsenal of stemmatologists. While in classical textual criticism such networks were not drawn and, technically, they are nowhere close to a tree, they have been adopted since they allow the testing of hypotheses involving contamination and are applicable to both open and closed traditions. For a visual rendering, see section 5.5.9. (On closed and open traditions respectively, see Spencer, Davidson, et al. 2004; Eagleton and Spencer 2006; see Bergel, Howe, and Windram 2016 for a discussion of this in the realm of print material.) Griffith (1984) already tried to depict visually how close manuscripts are to each other in a two-dimensional grid. A stemma is thus not the only way to display manuscript similarities, but if the goal is to approach the original text, it will still be the most effective tool.

5.1.2.3 Phylogenetic methods: Criticism, adaptation, adaptability

Stemma generation by phylogenetic software exhibits some graphical and some mathematical properties that render it incommensurate with previous methods used to produce graphs in textual criticism (see 5.5 for a fuller account). Most notable is the fact that phylogenetic software operates on DNA and protein code represented as pure string sequences. The software essentially understands DNA and proteins as merely linear sequences, even though there are non-linear dependencies between these natural phenomena which we do not understand very well. Likewise, language exhibits many layers of interdependence and structure, but phylogenetic software processes it as if it were a mere string sequence. Apart from this, the three most important different conceptual properties are the focus on leaves, bifurcativity, and unrootedness. Historically, these “alien” properties have been noted as technical challenges to be overcome. In the case of the focus on leaves, this has been achieved by Roos and Zou (2011); another bioinformatic method where the extant nodes can be non-leaves is described in Papamichail et al. (2017).

Bifurcativity in phylogenetic trees is the result of the most commonly applied concept of speciation (see, for more detail, Purves et al. 2004, 482; Hoelzer and Melnick 1994). Most algorithms of computational bioinformatics produce exclusively bifurcating trees; therefore, Bédier’s debate (see 2.3.4) is largely irrelevant in phylogeny. Multifurcating trees can be generated automatically by collapsing some of the bifurcations. In this case, those bifurcations are collapsed which according to some statistical method appear the least probable/reliable. Therefore, computational procedures that introduce multifurcation come at the cost of introducing one more parameter, a threshold for reliability. Usually, these procedures prevent unifurcations (representing the filiation of only one new witness) arising from collapsing bifurcations. Unifurcations may result, however, from manual intervention by the philologist. Philological trees generated by Semstem, which supports multifurcation, can also contain unifurcations.

Finally, as for the aspect of unrootedness, some automatic rooting approaches have been developed (Haigh 1970, 1971; Marmerola et al. 2016; Hoenen 2019). These approaches have been successfully tested on the limited testbed of artificial traditions. But they are still far from being practically used (see 5.4 below). Furthermore, programs can be used for the automatic generation of an archetypal text (see Hoenen 2015b; Koppel, Michaely, and Tal 2016), whose importance in stemmatology is another point of difference to the biological field. However, such methods are still very experimental and remain as yet little tested.

5.1.3 Recent developments

Around the beginning of the twenty-first century, two important books were published with collections of articles that were mostly concerned with computer-assisted stemmatology: the two volumes of *Studies in Stemmatology* (van Reenen and van Mulken 1996; van Reenen, den Hollander, and van Mulken 2004). Roos, Heikkilä, and Myllymäki (2006) invented an algorithm to compute a stemma which was not a direct loan from phylogenetics but designed to meet needs specific to stemmatology. Other approaches than phylogenetic ones have become more numerous in recent times; see, for instance, Roelli and Bachmann (2010); Roos and Zou (2011); and Lai, Roos, and O'Sullivan (2010).

Another important innovation has occurred: the first digital artificial traditions have been made. Datasets were produced by volunteers copying texts while the true stemmatic relationships were recorded and are thus fully known. These traditions can serve as test data for computational methods (see Spencer, Davidson, et al. 2004; Baret, Macé, and Robinson 2006; Roos and Heikkilä 2009). The first attempts to produce artificial datasets for philology go back to long before the digital era, when Kantorowicz carried out experiments from 1914 onwards. He had students copy texts, calling this new field “experimental textual criticism” (Kantorowicz 1921, 47; see Kleinlogel 1979, 64). However, only some reflections on the process and no results were ever published. Kantorowicz mentions (1921, 49) that, when trying to reconstruct the archetypal text manually from his students’ copies, he had judged roughly 10 % of the words wrongly because copyists had independently made the same changes. Apparently, this strand of research then disappeared from academic memory for a long time. Recently, however, such traditions have become useful as benchmarks for evaluation. Such artificial traditions have been obtained by having volunteers copy an actual original text from one another in a fixed and recorded sequence, in several copy-rounds, and finally digitising all the manual copies. Evaluation now means the comparison of a stemma computed from the “witnesses” with the recorded true stemma. This allows for an estimate of how well a method works and for the quantitative comparison of different methods. In computer science, such validation is an essential and integral part of methodology,

and therefore the availability of concrete test data invites more contributions from computer science. Roos and Heikkilä (2009) presented the results of a challenge that compared more than ten different algorithms in computing stemmata on the basis of three different datasets. This is, to date, the largest comparative study on algorithms for computer-assisted stemmatology. The reduced datasets for the traditions used there had almost no extant internal nodes, presumably in order to make the results more comparable – or comparable at all – with the output of bioinformatic programs. The evaluation was also designed in such a way that rooting or direction were not required.

Among the three disciplines producing “historical trees” (linguistics, biology, and stemmatology; O’Hara 1996), textual criticism is probably the domain that can attract contributions from computer science most easily, since, at least in case of closed traditions such as, for instance, the artificial *Parzival* tradition created by Spencer, Davidson, et. al. (2004), gold standards (that is, datasets for which the truth is known) are more or less easily produced. However, it should be noted that computer science is not exclusively evaluation-driven, but it is certainly the case that studies involving numerical evaluation are numerous and important in the field.

The physiological root of miscopying is by and large the same today as in Antiquity or the Middle Ages in the sense that the cognitive apparatus of humans then and now is supposedly the same. However, many phenomena, such as *scriptio continua*, the widespread use of abbreviations, and writing being less standardised, have not yet been taken into account in the artificial datasets. Also, many historically relevant writing systems (e.g. ancient Greek, Hebrew), which may have different miscopying characteristics, have not been targeted yet. It is also rather questionable whether artificial traditions will be able to simulate a realistic time-depth, and with that the influence of language change that is exhibited in historical data. Finally, intentional change – which certainly occurred in the copying of many texts – may be hard to model, as well as the interactions between oral tradition and written tradition. These caveats were realised early on and are sometimes named as a reason to reject using artificial datasets, as Poole stated:

To test the program [his Algol60 program] experimentally, it was first intended to run it with artificially constructed sets of data, incorporating deliberate contaminations. It soon became apparent, however, that these could hardly match the complexity of a real manuscript tradition, or put the program to a sufficiently rigorous test. (Poole 1974, 212)

Nevertheless, experiments on ever-new ways to produce artificial data or simulate text copying continue to be published (see e.g. Pompei, Loreto, and Tria 2018).

In summary, new digital artificial traditions are a new development in computational textual criticism that has parallels in the history of philology (Kantorowicz 1921). Although these artificial traditions certainly have limitations, they may make the field more attractive for computer science. Additionally, they may help to gain new insights into phenomena related to miscopying, their natural frequencies, and

similar properties, since they are based on the same (or similar) human cognitive apparatus that brought forth historical traditions. Finally, artificial traditions may be able to help us understand where manual philological habits are most probable to misinterpret certain aspects of historical data (see Andrews 2014).

5.1.4 The Bible and other very large traditions

Using the aid of the computer for such a large and complex tradition as the Bible has seen the need for special approaches. The number of required witness comparisons grows rapidly with corpus size, and this puts limits on the size of traditions that can be processed computationally with standard methods in feasible computing time. Wachtel (2004) developed a method called the Coherence-Based Genealogical Method (CBGM), in which a “textual flow” based on variant stemmata (see 4.2.3.6) classifies witnesses in terms of relative age (i.e. age relative to the other texts). This pre-genealogical coherence that is thus established is then used to generate a final stemma in a less computationally intensive way. This method can be applied to any tradition but will be especially useful in dealing with larger bodies of manuscripts where a larger number of variant trees can be produced. Another way of coping with very large traditions is to produce partial stemmata. Partial stemmata have been produced for different parts of the Bible. Lin (2016) summarises to some degree the history and application of computer-assisted stemmatology (as a sub-branch of computer-assisted textual criticism) in relation to the New Testament and beyond. On Biblical textual criticism, see further section 7.1.

5.1.5 The digital age and its contributions

We are certainly still in a phase of transformation from the print age to a digital era. As soon as digital and print paradigms start to mutually influence each other, inspiring print publications modelled on natively digital output, parallel to discussions following McLuhan’s (1962) famous *Gutenberg Galaxy*, we will be able to argue that we have moved into a “post-digital” paradigm. Some effects of the use of digital methods in stemmatology already point in this direction. The use of colour, for instance, is expensive in print, and mainstream stemmatic depictions in philology in the print age have rarely made use of this visual dimension. In contrast, since colour is cost-free (or at least cost-equal as compared to black and white) in the digital medium, stemmata and stemmatic software seem to have started to use colour in visualisations more widely. For instance, colour is used for group emphasis in C. J. Howe et al. (2001). Moreover, in the stemmata of Stemmaweb, colour is used in a variety of ways, such as showing which variants go with the stemma and which go against it. Another property of digital media is their dynamic nature. A stemma can,

for instance, be built up step by step, node by node, or it can be made zoomable. Eventually, even the underlying transcription might be changed on the fly. Other visualisation forms that are more easily produced using digital media are, for instance, three-dimensional stemmata and heat-maps with cladograms on top. This dynamic nature also implies many further possibilities, such as linking entities in the stemma directly to their textual content and to facsimiles and vice versa, zooming in and out, giving additional information through mouse tooltips, generating a manipulatable 3D stemma, using a time series to show different states of the stemma (or *arbre réel*, or both), and so forth.

Thus, the digital environment enables philology to develop a much richer visual language than that possible within the constraints of print technology. It is, however, still unclear which devices and visualisations will ultimately form part of a stemmatic methodological canon. For the time being, conventions for a new digital visual language for stemmatology are still being sought, and in that process we may very well see some further emancipation from print-age constraints.

Another advantage of the dynamic nature of digital media is that it facilitates the application of various concurrent models to the same input with a simple button click. This enables the textual critic to investigate the effects of assumptions and of different models in much more detail and with a better overview. Examining in this way a multitude of underlying models allows a more holistic stemmatic analysis. We can point to teicat.huma-num.fr as an example of such dynamics. This software allows the user, by pointing and clicking, to change which witness text will be the base text for a comparative apparatus. But, of course, the user should be aware of the consequences of changing parameters and base versions; thus, with the advantage also comes a larger responsibility and demand for some kind of digital literacy.

The possibilities these dynamics create may be seen as both a blessing (for offering more solid insights) and a curse (they likely involve more effort). Realising these possibilities also entails that the way in which stemmatologists (will) work in the digital age may change considerably in comparison to earlier times. The technical development of software and online publications involves more teamwork. Indeed, there is already a marked difference noticeable insofar as many publications from digital stemmatology, in contrast to classical stemmatology, are multi-author works. Even if they may require more work, for the digitally literate user, digital tools and visualisations encompass more possibilities to analyse and compare different stemmata given the same underlying text data. Because some sensible defaults can also be implemented strictly, this freedom does not have to result in less guidance. Finally, the debate on philology initiated by Bédier need no longer be of an existential magnitude, since any electronic edition can offer both a stemmatological approach and a best-text approach and leave the choice (or comparison) between the two up to the user.

Printing costs always have to be covered by someone, but – contrary to some popular but misguided beliefs – digital tools and data, and the access to and avail-

ability of them, are certainly not free (or cheaper) either. Thus, economic dimensions and considerations in stemmatological research projects remain. The digital environment also introduces some new challenges: the problem of the long-term availability of vulnerable digital resources, graphical interfaces rampant with visual overcrowding and ambiguity, quotability and versioning, user-friendliness and usability problems, potential “black-boxing” of methods, and so forth. All of these (potential) problems will have to be dealt with and examined to find ways to circumvent, mitigate, or solve them.

For stemmatology, the digital age is not only an opportunity to apply novel methods to arrive at a stemma, but also a chance to develop a richer visual language, to allow more exploration and dynamic devices, to compare different approaches and their implications. For those who are able to use multiple stemmatological approaches, this may enable deeper reflections, and a higher quality and consistency of results. It also certainly increases the complexity of the stemmatological endeavour considerably.

5.2 Terminology and methods

Sara Manafzadeh and Yannick M. Staedler

This section serves to provide a conceptual understanding of the mathematical objects, concepts, and methods involved with visualising and computing phylogenetic stemmata. For those who want to investigate the mathematical underpinnings more fundamentally, suggestions for further reading are provided.

5.2.1 The basic building blocks of graph theory

In graph theory, a node or a vertex (plural “vertices”) is the fundamental unit of which graphs are formed. An undirected graph is composed of a set of nodes and a set of edges (unordered pairs of nodes; see fig. 5.2-1a), whereas a directed graph is composed of a set of nodes and a set of directed edges (ordered pairs of nodes; see fig. 5.2-1b). Nodes are treated as featureless and indivisible objects, although they may have additional structure depending on the application. The two nodes forming an edge are said to be the endpoints of that edge. An edge connecting A to B is often written (A, B) , in which case nodes A and B are said to be adjacent. The neighbourhood of node A is the subgraph formed by all nodes adjacent to A , in other words all the nodes in the direct vicinity of node A . The degree of a node is the number of edges connecting to it; in figure 5.2-1a, nodes B and C have degree 4; nodes A , E , F , and G have degree 2; and nodes D and H have degree 1. A leaf node is a node with degree 1 (nodes D and H in fig. 5.2-1b). A graph is said to be connected if there is a path between any two nodes. A cycle is a

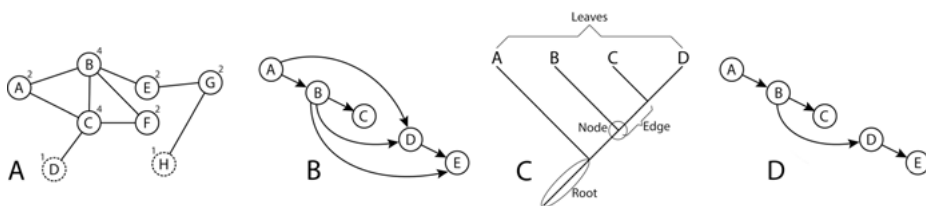


Fig. 5.2-1: Examples of graphs and trees.

path of nodes and edges in which a node can be reached again after setting out from it. More details about these mathematical entities can be found in Diestel (2005).

A directed acyclic graph (DAG) is a directed graph with no directed cycles (i.e. loops). It consists of a number of nodes and edges, with each edge directed from one node to another, in such a way that it is not possible to start at any node and follow a sequence of edges that loops back to the same node (fig. 5.2-1b). An undirected graph in which any two nodes are connected by one and only one path is called a tree. A polytree (or directed or oriented tree) is a DAG whose underlying undirected graph is a tree (fig. 5.2-1d is an example); stemmata representing the genealogy of a text without contamination usually take the shape of a polytree. In our contexts, trees are used to describe the genesis of related objects that evolve without interfering with one another. A phylogenetic tree represents the evolutionary ancestry of a set of tips. In biology, the tips (i.e. leaves; see fig. 5.2-1c) are usually extant species or groups of extant species (taxa, singular “taxon”), whereas in stemmatology tips represent extant witnesses of a text. Currently, almost all phylogenetic methods produce strictly bifurcating trees (also called binary trees) in which each node has at most two descendant nodes (see 5.1.2.1).

Looking more closely at the examples of graphs and trees may make things clearer. In figure 5.2-1a, an undirected graph is depicted. The circled letters in this graph are the nodes; the lines connecting the nodes are the edges. Here, the numbers above the nodes indicate the degree of each node, that is, the number of connections it has to other nodes. Letters encircled with dashed lines (nodes *D* and *H*) are leaf nodes, which like the leaves of a biological tree have only a single connection to the rest of the structure. Figure 5.2-1b is an example of a directed acyclic graph. In this type of graph, the relations between the nodes (i.e. the edges) have a direction (hence “directed”), indicated with arrows. Figure 5.2-1c shows the parts of a tree that are indicated with specifically phylogenetic terms.

A rooted tree is a tree in which one node has been designated as the root. Mathematically, this designation is arbitrary because the root can occur anywhere in the tree. People new to graph theory often have trouble with this concept. It helps to imagine the graph as a net, a set of beads connected with wires. If one were to lift the net by any randomly chosen bead, that bead (node) would become the root node, with all the other beads and wires hanging down from it. The edges of a

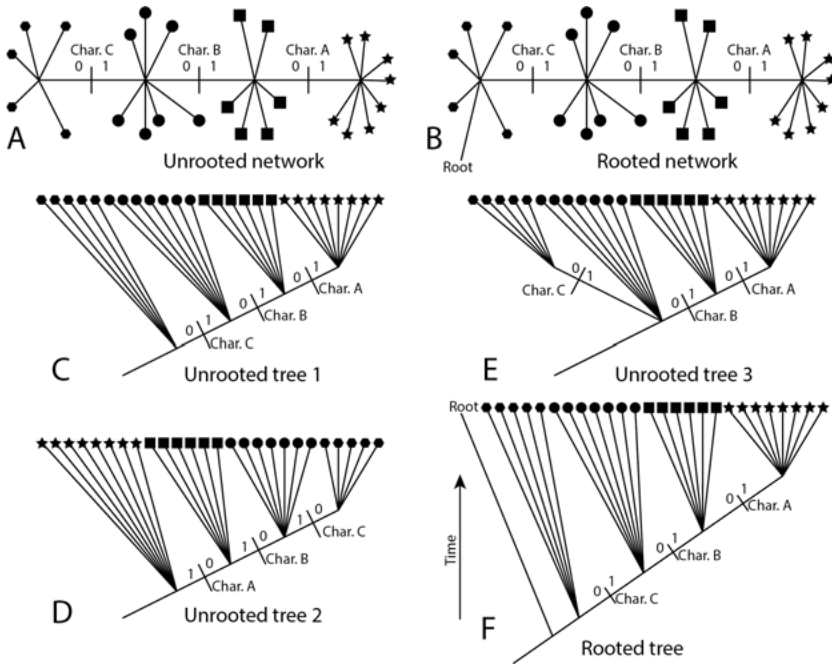


Fig. 5.2-2: Roots and their relevance for trees and networks. (a) is an unrooted graph (network). (b) is the same network as (a) but with a designated root. (c), (d), and (e) are simply different graphical representations of the same unrooted tree. (f) represents the same tree as (c), (d), and (e), but with a root.

rooted tree can be assigned an orientation, either all of them towards or all of them away from the root, in which case the structure becomes a directed rooted tree. If in a directed rooted tree there is one unique directed path to any node from the root, such a tree is called an *arborescence*. In phylogenetics, unrooted trees only display the relatedness of the leaves and do not represent a hypothesis of ancestry (cf. fig. 5.2-2a). Unrooted trees (see e.g. figs 5.2-2c–d) can always be generated from rooted trees (e.g. fig. 5.2-2f) simply by omitting the root. An unrooted tree is also called a *network*. Figures 5.2-2c–e show three different ways of recording and organising the same observations. Even though the network (fig. 5.2-2a) looks like a timeline, it is not: it could be read from left to right, from right to left, or from the middle outwards. To transform the network into a rooted tree, one must determine which changes are more recent than others; that is, the tree must be rooted.

A rooted phylogenetic tree is a directed tree with a unique node corresponding to the most recent common ancestor (MRCA) of all the entities at the leaves of the tree. In other words, that node represents a species from which all the other species in the tree eventually developed. Rooting polarises the character changes, giving them a direction. Again, if you imagine that the network is a piece of string, you can keep the connections exactly the same, even if you lift it up

in different places. The network from figure 5.2-2a is redrawn in figure 5.2-2b with the addition of a root. Different placements of the root can change the order in which the character changes occur in the tree.

Rooting a phylogenetic tree is critical for interpreting how taxa evolved, that is, how the various groups of species, or “leaves”, that are distinguished evolved. Different rootings suggest different patterns of change (i.e. different character polarisations). Of course, the pivotal issue is how the position of the root is to be determined.

The most common method for rooting trees in biology is by using an *outgroup*. An outgroup is a taxon that is a relative of the group under study. The key point of an outgroup is that, although related to the taxa under study, the outgroup taxon lacks some biological traits that are common to the group under study. Ideally, the outgroup should be close enough to allow inference from trait data or molecular sequencing, but distant enough to be a clear outgroup. For instance, the macaque can serve as outgroup for a group of apes under study. Clearly, macaques are related to apes, but apes, among other things, all lack a tail. On this basis, we can assume that macaques split off before any of the apes diverged as separate species. Thus, when selecting an outgroup, one must assume that all ingroup members (that is, the members of the group under study) are more closely related to one another than to the outgroup; in other words, the outgroup must have separated from the ingroup lineage before the ingroup diversified. Often, more than one outgroup is used to enhance the reliability of the hypothesis. If an outgroup is added to a network, the point at which it attaches is determined as the root of the tree. An analogue to biogenetic outgroups may be found in traditions that incorporate texts or parts of texts from other traditions – a text may have been included in a compendium or florilegium that has its own tradition, for example. Also, the existence of early translations (see the example in 4.5.3) can be seen as an analogue to outgroups. Usually, however, for stemmatologists no outgroups are available, and they have to turn to other methods to determine the roots in their trees (see 2.2), which are called *archetypes* (see 4.1.5).

Reticulation events are events that cause a new species to arise from the “merging” of two different parent species (note the difference to “normal” evolution, where traits of one species change over time until the species develops into a true new species of its own). An example of a reticulation event is *hybridisation*, where two species interbreed to produce a new (hybrid) species. Another example in biological evolution is *horizontal gene transfer*, where DNA migrates from one species to another. It happens, for instance, that bacterial DNA is moved from one bacterium species to another by a plasmid or a virus. Typically similar to reticulation events in stemmatology is *contamination*, as when a scribe used more than one exemplar to compile his copy (see 4.4). Such events are represented by *phylogenetic networks*. A phylogenetic network, or *reticulation*, is a graph used to visualise evolutionary relationships when reticulation events are believed to be

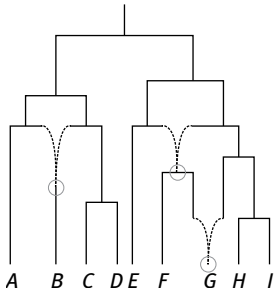


Fig. 5.2-3: A reticulated tree. Circled nodes represent nodes with two parents, also called hybrid nodes; dashed lines represent branches leading to a hybrid node.

involved (phylogenetic trees are a subset of phylogenetic networks). Whereas evolutionary trees usually only contain tree nodes, which are nodes with only one parent (see the continuous lines in fig. 5.2-3), reticulations contain additional hybrid nodes, which are nodes with two parents (marked with circles and dashed lines for lineage in fig. 5.2-3). An extension to the Newick format (see 3.4.8) is available for representing reticulations (see below; and Cardona, Rosselló, and Valiente 2008).

5.2.2 Phylogenetic inference

The data that is used to estimate the phylogeny of a set of leaves determines the characteristics of those leaves (taxa). The success of phylogenetic inference therefore depends largely on the choice of trait data and its accuracy and quantity. The first step in a phylogenetic analysis is to choose the taxa. The next step is to collect information on the traits of those taxa. These traits or properties are then stored in a data matrix. Two types of data matrices are mostly employed for carrying out phylogenetic analyses: character-state matrices or distance matrices. The character-state matrix can be viewed as a data sheet that has a list of taxa for the row headings. The columns represent properties or traits of species. Usually, each column is designated with a single character (with different possible character states). A character-state matrix has one specific entry for each character scored for each taxon (see fig. 5.2-4a). In this manner, a row with numbers becomes a very specific state description for a particular taxon.

In contrast, a distance matrix records for all pairs of taxa how dissimilar they are (or, more rarely, how similar). It therefore lists taxa both as row and column headers. The simplest way to compute a distance matrix from a character-state matrix is to calculate the proportion of characters for which two taxa differ in state in the character-state matrix. This value is then inserted into the appropriate cell in the distance matrix. It may be that a relative weight will be assigned to different characters, expressing the fact that we think (or have found) that certain traits or properties of species carry more information about genealogical relatedness than others. Returning to the macaques and apes example, we assume that having a tail is a very strong indicator that a species is not an ape. To express our very strong

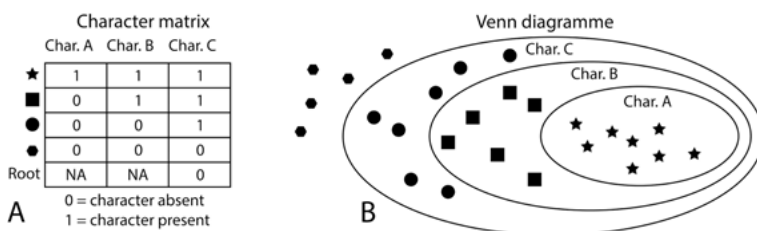


Fig. 5.2-4: Characters and trees. (a) character matrix; (b) Venn diagram derived from character matrix in (a).

suspicion, we can weight this trait by counting it more than once each time we find it in a species, and use this weighted value to calculate its distance from another species (thus setting it decisively further apart from species not having this trait). In stemmatology, the weighting of traits is connected to the issue of whether all “errors” in texts are equally revealing of genealogical relationships. Because opinions differ greatly about the relative weight of different variants, this has led to the study of significant errors (see 2.2.5).

Phylogenetic inference is based on the variable traits that have been scored for a set of taxa and that have been entered into one of the above-mentioned data matrices. A subsequent phylogenetic inference can be displayed in a number of different formats, for example tree diagrams (fig. 5.2-1c), hierarchical Venn diagrams (fig. 5.2-4b), indented classifications, Newick tree descriptions, or NEXUS tree descriptions. The two latter file formats are the most common formats for representing phylogenetic trees. The Newick format was created to represent trees in a computer-readable form. The development of the NEXUS format, which extended this format to encapsulate additional phylogenetic data, began in 1987 (Maddison, Swofford, and Maddison 1997). NEXUS files applying the Newick format are the most commonly used way of representing tree topologies through the use of characters (instead of visual lines, boxes, circles, and so forth). Monophyletic clades, that is, species or groups that share one common ancestor, are surrounded by parentheses, and sister clades are separated by commas. As an example, the tree in figure 5.2-1c can be written in Newick format as $((C,D),B),A$). The Newick format can also contain additional information about branch lengths (after colons) and node names (after closed parentheses). Each NEXUS file contains the following basic blocks: a data block containing the data matrix, a taxa block containing information about taxa, and a tree block presenting phylogenetic trees in Newick format (Archie et al. 1986). On this format and for an example, we refer the reader to figure 3.4-27 above.

5.2.3 On distance measures

In phylogenetic tree-construction methods, one will often encounter mentions of “distance” in connection to character sequences, matrix columns or rows, texts, and

so forth. Such distance measures quantify how similar or dissimilar strings of characters are – note that character sequences, matrix columns, matrix rows, and texts can all essentially be understood as strings (or rows) of characters. The sheer number of different distance measures that have been developed defies any exhaustive listing; we will therefore, for the sake of clarity, present here only a few, basic approaches. Distance measures broadly fall into two categories: edit distance measures and vector distance measures. Edit distance measures express the difference between character sequences based on the minimum number of mutations that are needed to turn one sequence into another. “cat”, for instance, is one edit distance away from “cot” (one substitution of “a” with “o” is required), and “cat” is two edits away from “cost” (one substitution of “a” with “o” and one addition of “s”). Many edit distance measures and related algorithms exist. Most notable and most frequently used are the Levenshtein distance and the Longest Common Subsequence (LCS). Levenshtein distance computes the minimal amount of insertions, deletions, and substitutions needed to morph one string into another. LCS, as its name suggests, calculates distances based on the longest coinciding substrings of characters it can find in the texts that are compared. For text distance measures in philological practice, it is advisable to apply a variant of Levenshtein, the Damerau–Levenshtein distance, which takes transpositions into account as a single edit. Vector-based distance measures express character or word sequences as paths in a high-dimensional space where each dimension represents the occurrence or frequency of individual words or characters in the sequence. The distance between sets of words or characters can then be computed as either the L1 distance (more colloquially known as Manhattan distance), which computes the number of steps that need to be travelled along every axis to reach another point in the high-dimensional space; or (more commonly) as the Euclidean distance; or as a cosine measure which computes the angle between two vectors. For a comprehensive overview of this topic, refer to Goma and Fahmy (2013).

5.2.4 Tree-reconstruction methods

Phylogenetic trees are usually inferred from genetic sequences or morphological data in biology, and from errors or variants in stemmatology (see 2.2). Phylogenetic reconstruction methods are based either on distance or on character data. In distance matrix methods, the distance between every pair of data sequences is calculated as explained in section 5.2.2. The distance matrix thus obtained is then used for tree reconstruction. Phylogenetic reconstruction methods based on character data include maximum parsimony, maximum likelihood, and Bayesian inference methods. Each method is introduced below and explained in more detail in section 5.3. For an in-depth treatment of the various concepts outlined here, we refer

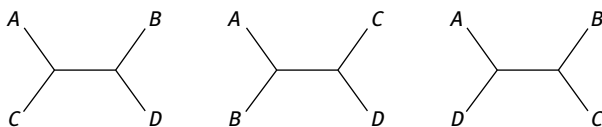


Fig. 5.2-5: All possible bifurcating unrooted trees when there are four taxa.

to Ziheng Yang and Bruce Rannala (2012). Here, we try to describe some common characteristics of these tree-construction methods and to provide some intimations about them.

The objective of any tree-reconstruction method is to infer the most likely tree given a number of taxa. It is theoretically possible to generate all possible trees for a given number of taxa. A possible way, therefore, to solve this problem is to simply draw all possible trees and subsequently compute which tree fits best according to some criterion. The problem with this approach is the enormous increase in possible trees as the number of taxa rises. For two taxa, there is really only one possible tree (two nodes and an edge). With four taxa, there are three possible unrooted bifurcating trees (see fig. 5.2-5). If we progress in this way, we can indeed draw all the possible trees for any number of taxa, but with five taxa the number of possible trees is 15, and with ten taxa it is already 2,027,025. With twenty taxa, the number of possible trees rises to some 222,000,000,000,000,000,000. Even for powerful modern computers, it is impossible to calculate that number of trees within a feasible time. Because the number of possible trees is so impossibly large, heuristic tree-search algorithms are used to bypass this problem. Heuristic approaches work by first generating a starting tree that fits the observed data using a rapid algorithm. After this, the algorithm tries to improve the score of the tree by making incremental changes to the tree based on heuristic data (hence “heuristic approach”). It should be noted that with heuristic methods there is always the possibility that the best tree may not be found because there may be many valid possibilities that are not computed.

The methods differ in the way they compute how well a tree fits the observed data, the *tree score*. In maximum parsimony-based methods, a tree score is considered better if fewer changes (mutations) are needed between taxa to realise the tree. In maximum likelihood methods, the tree score is based on the log-likelihood, which can be understood as a value that indicates how probable a tree is given the model for mutations the method uses. For Bayesian inference, the score is also a probability value, the posterior probability which is influenced by actual observed data. Maximum likelihood and Bayesian inference use a model for data change, that is, they assume in their calculations that mutations occur according to a certain given mechanism. They assume, for instance, that mutations are always DNA base substitutions or text modifications. Maximum parsimony does not have such an explicit model; it computes a score purely on the basis of the number of changes and does not adapt its approach based on the type of change.

In principle, all tree-building methods face the problem that it is only possible to generate all possible trees for a small number of taxa and that, for larger numbers of taxa, “short-cut” approaches need to be used. It is therefore always possible that the solution computed is actually not the real solution. To mitigate this problem to a certain extent, a technique called *bootstrapping* is often applied. A bootstrap method consists of running the tree-building process a set number of times (e.g. 100 times), inputting the same data in a different (e.g. randomised) order or using different samples, or both. The consistency of the position of nodes and branch lengths can then be computed across all results as a ratio of how many times the same nodes and branch lengths appear in the calculated results. The larger this ratio, the more reliable the result is assumed to be.

5.2.4.1 Parsimony

Pluralitas non est ponenda sine necessitate. (Ockham 1967, 74)

The principle of parsimony, or “Ockham’s Razor”, named after Guillelmus de Ockham, OFM (1285–1349), states that when trying to explain a phenomenon, it is better to prefer the explanation involving fewer assumptions. In evolutionary biology and in stemmatology, the principle of parsimony is relevant because it is assumed that the probability of the same mutations (or text alterations) evolving independently is low. Empirical evidence strongly suggests that biological evolution is primarily based on random mutations in DNA. After a mutation, the altered DNA is propagated in a species through offspring. For this reason, it is assumed that if two species carry the same alteration in their DNA, it is highly unlikely that these alterations arose independently, and instead it is assumed that both species have a common ancestor species in which the change happened at some point. Similarly, in manuscript evolution it is assumed that if two witnesses have the same variant reading, the cause is much more likely to be that a common ancestor had that same reading, and not that the variant occurred twice independently. It should be noted, however, that this is only statistically true: in biology, because there are only four possible DNA bases, identical changes that happen independently do occur; they are just much less likely.

Parsimony tree score

The maximum parsimony tree is the tree for which the tree score is lowest. The way maximum parsimony minimises the number of changes in a tree is by assigning character states to the interior nodes of the tree in such a way that the changes in character states from node to node are minimal. The particular place of a character in a character state is called the *site* of that character (roughly corresponding to the *loci critici* in textual criticism; see 3.3.4), and a mutation means that a site becomes occupied by a different character. Obviously, there is a minimum number of

changes that are required to progress from one character state to the next. This number is called the character length or site length. The tree score is the sum of the character lengths across all sites.

Some sites are not useful for parsimony-based tree comparison. Constant sites are sites for which the same nucleotide or text string occurs in all species or witnesses; they thus have a character length of zero in any tree, and are parsimony uninformative. Singleton sites are sites at which only one of the species or manuscripts has a distinct nucleotide or text string whereas all others are the same. These singleton sites can also be ignored because they do not allow a common ancestor to be inferred, which would require another species to also have that particular nucleotide or text string at that site. In stemmatology, these singleton sites are known as *Eigenfehler*, *Sonderfehler*, or *lectiones singulares*. The parsimony informative sites are those at which at least two distinct characters are observed, each at least twice.

A controversy arose in the 1990s as to whether maximum parsimony (without explicit assumptions) or maximum likelihood (with an explicit evolutionary model) was the better method for phylogenetic analysis. Today, the importance of model-based inference methods is broadly recognised. Parsimony, however, is still commonly used. Not because it is believed to be free of assumptions, but because it is computationally efficient and often produces acceptable results.

Strengths and weaknesses of parsimony

Parsimony's strength is its relative simplicity, which makes it easier to describe and understand. Moreover, it is amenable to rigorous mathematical analysis. Parsimony's primary weakness is its lack of explicit assumptions. This makes it almost impossible to include any knowledge about the process of sequence or text evolution to be applied during the tree-reconstruction process. Parsimony's failure to correct for multiple substitutions at the same site causes a problem known as long-branch attraction: if the correct tree has two long branches separated by a short branch, parsimony will tend to group the long branches together. In such cases, parsimony converges on a tree that is wrong. It should be noted that model-based methods (maximum likelihood and Bayesian methods) also suffer from long-branch attraction if the sequence or text evolution model is too simplistic and ignores, for instance, the rate of variation across sites.

5.2.4.2 Maximum likelihood methods

Maximum likelihood is a statistical method developed to estimate unknown parameters in a model. To understand what this means, we can, for instance, suppose that the monetary value of a painting is dependent on the surface area of its rectangular canvas – which may be less naive an assumption than one would think (see Renneboog and van Houtte 2002, 339). A model in that case may be $v = l \times b$,

meaning that the value equals the length times the breadth of the canvas. But suppose also that we observe that this is not entirely correct: the larger the painting, the more the value estimated by our model falls short of the actual value. If that is the case, a better model would be $v = \alpha \times (l \times b)$. In that model, α is a parameter. The question in this case is: what is the value of that α ? This is typically the sort of problem where maximum likelihood is applied. If we have prices and actual measurements of paintings, we can compute the likelihood of the observed data for any given value of the parameter α . The maximum likelihood estimate (MLE) of the parameter, then, is that value of the parameter which maximises the likelihood of observing the actual data in real life. In tree-reconstruction calculations, many such parameters may have to be estimated. Such unknown MLEs are usually assessed numerically via iterative optimisation algorithms.

Maximum likelihood tree reconstruction

Owing to increased computing power and advances in software implementation, and to the development of increasingly realistic models of sequence evolution, this method is now widely used. Maximum likelihood tree estimation involves two optimisation steps: (i) the optimisation of branch lengths to calculate the tree score for each tree, and (ii) a search in tree space for the tree with maximum likelihood. The tree (topology) is, from a statistical point of view, a model (see 4.2). Branch lengths in the given tree and substitution parameters, on the other hand, are parameters in the model. Maximum likelihood inference is therefore equivalent to a comparison of many statistical models with the same number of parameters. Most models used in molecular phylogenetics assume that the sites in the genetic sequence (or the text modifications) evolve independently: the likelihood is therefore a product of the probabilities for the different sites. The probability at any given site is an average over the unobserved character states at the ancestral nodes. Parsimony and likelihood analyses are similar in this aspect, although parsimony uses the optimal ancestral states only, whereas likelihood averages over all the possible states.

Strengths and weaknesses of the maximum likelihood method

The maximum likelihood method has two major advantages. The first is that all of its model assumptions are explicit and can therefore be evaluated and improved. Second, a broad range of sophisticated evolutionary models is available for likelihood-based methods. If the aim is to understand the process of witness or DNA sequence evolution, the maximum likelihood method has clear advantages over the minimal parsimony approach. The main disadvantage of maximum likelihood is that the likelihood calculation, and the tree search in particular, are computationally intensive. The other drawback of the method is that false or too simple models can be inaccurate about tree reliability, that is, they can suggest that the estimated tree is significantly supported when it actually is not (Z. Yang, Goldman, and Friday 1994).

5.2.4.3 Bayesian phylogenetics

The difference between Bayesian inference and maximum likelihood-based methods is that parameters in a Bayesian model are random variables with statistical distributions, whereas in maximum likelihood-based methods they are unknown constants. In other words, the Bayesian variant of our very simple $v = \alpha \times (l \times b)$ model for the value of paintings assumes that we should not compute the value of the parameter α as a fixed value (e.g. as exactly 5 or 0.4). Instead, the Bayesian variant of that model asserts that α may vary between certain values. The Bayesian model, in this manner, calculates what the likelihood of values for the parameter α is. In real-world situations, such parameters are assigned a “prior distribution” before the data analysis (i.e. the likelihood of the minimum and maximum values of the parameters is chosen or given, for instance based on earlier experience). This prior distribution is combined with actual data to generate a posterior distribution, and final parameter inferences are then based on this posterior distribution. Bayesian inference relies on Bayes’s theorem:

$$P(T, \theta | D) = P(T, \theta) \times P(D | T, \theta) / P(D)$$

where

$P(T, \theta | D)$ is the posterior probability,

$P(T, \theta)$ is the prior probability for a tree T and a parameter θ ,

$P(D | T, \theta)$ is the likelihood or probability of the data given the tree and parameter, and

$P(D)$ is a normalising constant to ensure that the sum over the trees and integration over the parameters of $P(T, \theta | D)$ is 1.

The theorem states that the posterior probability is proportional to the prior probability multiplied by the likelihood of the data given the parameters. Most often, the posterior probabilities of trees cannot be calculated directly, and calculating the normalising constant $P(D)$ is especially arduous. Bayesian inference therefore relies on Markov Chain Monte Carlo algorithms to create a sample from the posterior distribution.

Strengths and weaknesses of Bayesian inference

Both likelihood-based methods and Bayesian methods use a likelihood function. Advantages and drawbacks that apply to likelihood-based methods apply, therefore, equally to Bayesian methods. Bayesian statistics answers the biological or stemmatological questions in a relatively straightforward manner because a tree’s posterior probability is simply the probability that the tree is correct, given the data and the model. In contrast, interpreting the confidence intervals in likelihood analyses is more complex: in phylogenetics, it has not been possible to define a confidence interval for trees, and the widely used bootstrap method is rather difficult to interpret.

On the other hand, Bayesian posterior probabilities for trees and clades calculated from real data often seem excessively high: in numerous analyses, most nodes have posterior probabilities of about 100 %. Posterior tree probabilities are sensitive to model violations, and the use of simplistic models may lead to inflated posterior probabilities. Moreover, although the prior probability allows for the incorporation of a priori information about the trees or parameters, such information is most often unavailable. Furthermore, high-dimensional priors are hard to specify, and they may influence the posterior probability in unexpected ways. Bayesian robustness analyses are therefore crucial for assessing the impact of the prior on the posterior estimates.

Despite these caveats, and thanks to advances in computational methods, Bayesian inference has increasingly gained in popularity in the past two decades.

5.3 Computational construction of trees

Teemu Roos

In this section, we outline the main approaches and methods for the automatic construction of hypotheses for genealogical trees. We caution the reader that these methods are to be applied as a part of a computer-*assisted* approach – instead of a “computerised” or fully automated approach – and that the results need to be subsequently critically examined and interpreted by the scholar. It is never a good idea to blindly accept whatever result these methods produce as the “correct” result. We avoid the use of the term “stemma” and use the term “tree” instead of it because, in our terminology, a stemma is a rooted diagram whereas the trees obtained by computer-assisted methods are almost invariably unrooted. Admittedly, the use of the term “tree” is also somewhat inaccurate, due to the fact that some of the methods actually produce networks rather than trees. Adopting the phylogenetic terminology, we refer to the objects whose relationships we are interested in as “taxa”, instead of “witnesses” or “manuscripts”, in this section.

5.3.1 Manual and computational construction

Traditionally, stemmata are constructed manually. They are based on a collation and careful scrutiny of the source material. It is noteworthy that manual, that is, non-computer-assisted stemma construction should follow a rigorous and strict procedure too. By this, we mean a procedure where each decision is based on sound principles applied to “internal” evidence in the collated material and possibly complemented by “external” evidence from other sources. Assuming that such a rigorous procedure exists, it follows that, in principle, the procedure can be formalised

as a set of explicit rules for constructing a stemma – or, in other words, an algorithm. However, while all that is true in principle, in practice it is a simplification. In particular, external evidence, which can be made use of in order to, for instance, understand the historical context in which the source material was created, can be extremely hard to formalise with a set of clear-cut rules. This is why we use the term “computer-assisted stemmatology” rather than, say, “automatic stemmatology” or “artificial intelligence stemmatology”.

In computer-assisted stemmatology, the working method often involves an iterative process where a hypothesis is constructed by an algorithm and scholars then reflect on the results by calling on their scholarly expertise (knowledge of the text, knowledge of the historical context, materiality, and so on). If the hypothesis is not entirely satisfactory, they may decide to adjust the method or the source material. This may lead to different encodings of the data, collating more material, removing some taxa (witnesses), splitting the material into multiple parts and analysing them separately, and so forth. Some methods may also include adjustable parameters or constraints that affect the outcome. Of course, it is pivotal to avoid the temptation to keep fiddling with the material or the method until a “desired” result is teased out of it. To this end, the iterative process should also follow clear and rigorous principles, and it should be documented carefully and disclosed together with the results obtained. We are not aware of a set of explicit principles of this kind, and we point this out here as a much-needed contribution to the field.

5.3.2 Classes of methods

In order to make it easier to get a grasp of the variety of approaches, we adopt the same categories or classes of methods as in section 5.2. Distance-based methods accept as input a set of pairwise distances between the taxa – that is, a list of distances according to some measure for each possible pair of taxa. The parsimony-based methods category mainly covers the maximum parsimony tree-construction method, where the objective is to minimise the number of “mutations” required to explain the variation in the data. Statistical methods are a large class of techniques based on statistical principles such as maximum likelihood. Bayesian methods are a subclass of statistical methods which we treat in a separate subsection. Finally, we describe methods that are specifically designed for stemmatological applications in their own subsection.

Of course, we can only ever provide an incomplete list of the existing methods in each subsection. The subsection categories or classes are also not mutually exclusive, and some methods might have been placed in one category just as well as in another. An example is the least-squares method, which we classify as a statistical method even though it is also a distance-based method.

For each of the described approaches or methods, we follow the same structure as far as possible:

- syntax and semantics of input data,
- key ideas,
- syntax and semantics of the output,
- underlying assumptions, and
- examples of application in the literature.

5.3.3 Distance-based methods

Distance-based methods operate on pairwise distances between the taxa. These can be obtained in different ways, which can obviously affect the outcome in significant ways. In stemmatology, a typical measure of distance is simply the number of words that are different. However, the treatment of changes in word order, gaps, non-words such as punctuation, annotations, colours, and other typographical elements needs to be decided. Section 2.2.5 describes how some changes are more relationship-revealing than others. Differences may therefore be weighted accordingly. This task is traditionally done on the basis of the expertise of a philologist. Nothing, however, hinders us from trying to define a model to handle this task automatically. Another potentially critical decision is whether to apply some sort of distance correction or not (see Spencer and Howe 2001).

5.3.3.1 Minimum spanning trees, arborescences, and Steiner trees

A minimum spanning tree is an undirected tree-shaped graph that connects a given set of nodes (taxa) by edges such that the total sum of the edge weights given by the pairwise distances between the corresponding nodes is minimised. Classical algorithms for constructing minimum spanning trees include Prim's algorithm and Kruskal's algorithms (see e.g. Cormen et al. 2009). The resulting tree diagram cannot usually be interpreted directly as a stemma because (a) it is unrooted and (b) it does not include any unobserved (missing) ancestral nodes at the branching points. Instead, each of the branching points is always occupied by a taxon corresponding to an extant version of the text. Related graph-theoretical concepts include arborescences (directed rooted trees) and Steiner trees (minimum spanning trees that allow additional nodes to be created to serve as branching points). However, these are rarely used in phylogenetics or stemmatology.

5.3.3.2 UPGMA

The Unweighted Pair Group Method with Arithmetic Mean (UPGMA) is a classical hierarchical clustering technique (Sokal and Michener 1958). The gen-

eral idea is to start with separate “clusters” for each taxon, and iteratively merge the most similar pair of clusters in each step. When two clusters are merged, they are removed from the set of clusters and replaced by a single new cluster. Eventually, there will be only two clusters left, which are merged in the last step of the algorithm. The order in which the clusters are merged produces a tree structure such that in the bottom level of the tree, we have pairs of taxa, and the higher levels of the tree correspond to the steps where clusters consisting of multiple taxa have been merged.

Many hierarchical clustering techniques exist. They differ from each other in terms of how the distance between the newly created cluster and the other clusters is defined. Let A, B, \dots, E be taxa and the pairwise distances be denoted by $d(A, B)$, $d(A, C)$, $d(B, C)$, and so on. Let us now assume that it turns out that $d(D, E)$ – thus the distance between taxa D and E – is the smallest of all the pairwise distances. Because of this, the algorithm will begin by merging D and E . (Note that, in the beginning, each taxon is its own “cluster”.) Then let us denote the new cluster by DE . We now need to define the distance between the new cluster and the other remaining clusters, A, B, C , and so forth. In UPGMA, this distance is defined as the arithmetic mean of the distances from the individual distances of A, B, C , and so on to the taxa D and E . So, for example, the distance $d(A, DE)$ is defined as $(d(A, D) + d(A, E)) / 2$. Moreover, when merging two clusters that consist of an unequal number of taxa, the two distances are weighted by the respective cluster sizes. So, for example, if we were to merge clusters A and DE , the distance $d(B, ADE)$ would be defined as $(1 \times d(B, A) + 2 \times d(B, DE)) / 3$, where the denominator 3 is the total number of taxa in the two merged clusters counted together. This process is now repeated until only one cluster remains.

The output of the algorithm is a rooted and directed tree structure. In addition to the topology of the tree, branch lengths are produced. The branch lengths are defined by the cluster-to-cluster distances when merging. We will not discuss the details of branch-length estimation, but, roughly speaking, short branch lengths indicate compact (more similar) clusters, while long branch lengths correspond to clearly separated clusters.

The UPGMA method can be shown to be consistent, that is, to produce the correct tree structure if one exists, under the assumption of a molecular clock. In technical terms, this is called the ultrametricity assumption. Intuitively, it means that all the lineages evolve at a constant rate and that the taxa are observed contemporaneously (at the same point of time). In terms of the tree structure, this implies that the leaf nodes (taxa) are at a constant distance from the root (most recent common ancestor, or archetype). This is usually not at all the case in text evolution. If (and when) this assumption is violated, the tree can be severely distorted. Another common problem scenario is the long-branch attraction phenomenon, where taxa that are very dissimilar to the others tend to cluster together even if they are also dissimilar to each other (Felsenstein 1978a).

5.3.3.3 Neighbour-joining

Neighbour-joining (NJ) is another commonly used distance-based method (Saitou and Nei 1987). Like UPGMA, it is also based on hierarchical clustering. There are two crucial differences related to the definition of pairwise distances. First, the pair of clusters to merge is selected by minimising an adjusted distance, Q , which is defined in a way that is designed to account for variable evolutionary rates. Intuitively, the long distances that are due to rapidly evolving lineages are discounted by subtracting the average distance between a taxon and the other taxa. Second, the definition of the cluster-to-cluster distances is adjusted in a similar fashion. The combined effect of these adjustments is that the method is not subject to long-branch attraction.

The input and the output of the method are similar in syntax and semantics to those of the UPGMA method, with the exception that the NJ tree is undirected. While the NJ method is not subject to long-branch attraction and does not require the ultrametricity assumption, it is still based on the assumption that the input distances faithfully reflect the genealogy. As with any distance-based methods, choices about the data encoding, treatment of gaps, and so on, as well as the use of distance correction, can make a significant difference to the outcome.

As tends to be the case with distance-based methods, NJ is relatively fast and scales up to hundreds of taxa. Furthermore, variants of the algorithm have been presented that can be applied to thousands of taxa (K. Howe, Bateman, and Durbin 2002).

5.3.4 Parsimony

Parsimony-based approaches are another classical category of phylogenetic methods. As opposed to distance-based methods, parsimony methods require a set of character sequences as input. The sequences must be aligned so that they can be easily compared character by character. The sequences are placed in the leaf nodes of a tree, where the internal (non-leaf) nodes correspond to ancestral taxa whose sequences are unobserved. If we attach hypothetical sequences to all the internal nodes, we can calculate for each edge in the tree a score which is simply the number of characters where the sequences at the opposite ends of the edge differ. The score of the whole tree, with the chosen hypothetical sequences, is then the sum of the scores of all the edges in it. The “small parsimony problem” is to find the set of hypothetical sequences that minimises this score for a given tree. The “large parsimony problem” is to find the tree *and* the hypothetical sequences at the internal nodes that minimise the score.

Computationally, the small parsimony problem is easy and can be solved in linear time with respect to the number of taxa by an elegant message-passing algorithm (Fitch 1971). This means that the calculation time increases linearly with the

increase in taxa: if it takes four minutes to solve the problem for four taxa, it will take eight minutes for eight taxa. However, the large parsimony problem is, in computer science terms, “hard”. The precise technical term is “NP-hard”, and it implies that no scalable algorithm for solving it exactly is believed to be possible. In practice, therefore, the only possibility is to use a heuristics-based search that does not guarantee an exact solution except for a very small number of taxa (about a dozen).

The output of the method is a tree structure. The edge-specific scores mentioned above can be used to define branch lengths, which have a similar interpretation to that of the branch lengths in distance-based methods: small branch lengths indicate compact groups of taxa, while long branches indicate clearly separated groups. The logic is quite straightforward: if a set of taxa differ from each other in only a few characters, they tend to be grouped together with small branches separating them, whereas groups of taxa that differ by many characters will also be far apart in terms of the parsimony tree.

The maximum parsimony method has been criticised for producing misleading results due to long-branch attraction (see 5.3.3.2) and other scenarios where the number of differences is not directly proportional to the evolutionary distance between the taxa (see e.g. Felsenstein 1978a). However, in practice it is still widely applied, and often its performance is found to be relatively good.

Parsimony has been used several times in stemmatology – for instance, Robinson and O’Hara (1996); Baret, Macé, and Robinson (2006); Roos and Heikkilä (2009); and Tehrani (2013).

5.3.5 Statistical methods

From a statistical point of view, we can consider the tree model as a parameter to be estimated from data. There are various ways in which this can be done.

5.3.5.1 Least squares

Possibly the simplest scenario is one where the tree topology (the structure of the tree) is fixed and we only need to estimate the branch lengths. If the input data is in the form of pairwise distances, the “fit” of the model can be defined by comparing the input distances, $d(u, v)$, for all pairs of taxa u and v to the “tree distances”. With “tree distance”, we mean the length obtained by adding up the branch lengths on the path from u to v . This problem can be converted into a linear regression problem where the branch-length parameters correspond to coefficients which can be estimated using the standard least-squares method. The goodness of fit is given by the sum of squared errors in the distances (observed vs tree distances). This is analogous to the small parsimony problem discussed above. The corresponding large problem is to find the tree topology for which the goodness of fit is the best (minimum sum of squared errors; Cavalli-Sforza and Edwards 1967). Similar to

the maximum parsimony method, the large problem is NP-hard, and no exact solution is guaranteed for more than about a dozen taxa.

Variants of the method exist where the distance errors are treated differently by, for example, weighting small distances more than large distances (see e.g. Fitch and Margoliash 1967).

The input data for the least-squares method is in the form of pairwise distances, so the method can also be categorised under distance-based methods. Consequently, all the considerations about the definition of distances, distance corrections, and so forth apply. The output is a tree with branch lengths. The interpretation of the branch lengths is also the same as in other distance-based methods.

The assumption underlying the least-squares method is – loosely speaking – that the distances reflect the evolutionary distance. Under this assumption, most variants of the method can be shown to be consistent (Rzhetsky and Nei 1992). In particular, least squares does *not* require the ultrametricity assumption (all lineages evolve at a constant rate), and it is not prone to the long-branch attraction problem.

5.3.5.2 Maximum likelihood

The statistical model underlying the least-squares method (see above) is not based on any concrete probabilistic model of sequence evolution: it simply assumes that the observed pairwise distances reflect evolutionary distances. Various explicit models of sequence evolution have been proposed in evolutionary biology. A sequence evolution model assigns a probability for a descendant sequence (e.g. CAGTA – A, C, G, and T denote the nucleotides in DNA sequences) to be produced from another, ancestral sequence (e.g. CAGAA). The model is typically parametrised by a branch-length parameter (or parameters) that corresponds to the time passed between the ancestral and the descendant sequences and an evolutionary rate at which mutations tend to occur per unit time.

Examples of sequence evolution models include the Jukes–Cantor model, often abbreviated as JC69 (Jukes and Cantor 1969), and the Kimura model (Kimura 1980), abbreviated as K80. The models have a varying number of parameters. For example, the JC69 model has only one parameter, which is the overall mutation rate. The K80 model, on the other hand, has two parameters to control the rate of change. One parameterises the A/G and C/T transitions. The other parameter pertains to the remaining mutations, A/C, A/T, G/C, and G/T. Similar sequence evolution models exist for protein sequences.

Again, we can separate the small phylogeny problem, which is estimating the parameters of the sequence evolution model for each branch under a given tree topology, and the large phylogeny problem, which is to find the tree topology as well as the parameters. In both cases, the maximum likelihood principle says that we should maximise the probability of the observed sequences under the model (tree and parameter values). In this case, even the small problem is computationally hard, and typically heuristic techniques based on the expectation maxi-

misation (EM) algorithm are applied. The large problem is, again, harder still, and again heuristic search algorithms are commonly used to find a good, but possibly not the best, topology.

The sequence evolution model makes the assumptions of the maximum likelihood model explicit: the sequences are assumed to have evolved independently along the lineages in the tree according to the chosen sequence evolution model.

The input of the maximum likelihood method is a set of aligned sequences for the extant taxa. The output is a tree topology with branch-length parameters. Since the models may have multiple parameters, the branches do not necessarily have only a single branch-length parameter. However, in most cases, one is singled out in order to be able to draw the trees.

5.3.5.3 PhyloDAG

PhyloDAG is an extension of the maximum likelihood approach in order to handle non-tree-like relationships (Nguyen and Roos 2015). It is based on an evolutionary model proposed earlier by Strimmer and Moulton (2000) that allows a descendant sequence to have two parents rather than only one. This implies that the model topology is defined by a directed acyclic graph (DAG) instead of a tree.

The main challenge and drawback of adopting the more general DAG model instead of trees is the computational cost. Finding a good DAG is an extremely slow process and works reliably only for small datasets with up to 20–30 extant taxa. For larger datasets, the search time becomes prohibitive and the quality of the results degrades as the heuristic search fails to find good solutions.

As in the case of the maximum likelihood method, the input of the PhyloDAG method is a set of aligned sequences. The output is a DAG where nodes correspond either to unobserved ancestral taxa or observed extant taxa. The most interesting property of the output is often the arrangement of “reticulations” (the nodes with two parents), if any. Since the heuristic search method used in PhyloDAG is not deterministic, it is advisable to repeat the analysis multiple times to obtain a range of possible solutions. The method also outputs a log-likelihood score which measures the goodness of fit. However, since the models may have a variable number of parameters, direct comparison of the log-likelihoods is not meaningful, and an additional comparison stage, such as bootstrapping, is recommended.

Currently, the only sequence evolution model available in the PhyloDAG software package is the JC69 model (see above). The reticulation model of Strimmer and Moulton (2000) makes the assumption that each character in a reticulation node is inherited from a parent that is chosen independently of the choices made for the other characters. This assumption may be quite unrealistic since the sequences are often inherited as longer segments, each of which is inherited from a single parent, instead of randomly switching between the parents at each position along the sequence.

Tehrani, Nguyen, and Roos (2016) apply PhyloDAG to resolving the genealogy of the fairy tale *Little Red Riding Hood*. They recommend a parametric bootstrap procedure for comparing a number of output DAGs.

5.3.6 Bayesian methods

The defining property of Bayesian methods is that they assume that models also have prior probabilities which, together with the observed data, determine the outcome. For example, we can assume that the branch lengths are distributed according to some probability distribution. Given data, we can compute the posterior probabilities of the branch-length values. The posterior probability of a parameter value (e.g. branch length = 1.5 units) is proportional to its prior probability multiplied by the likelihood (i.e. probability) of the data given the value. Thus, the posterior probability is highest for parameters that have high prior probability and which explain the data well (high likelihood). The posterior probability is obtained from the prior probability and the likelihood by the Bayes rule (Bayes's theorem; see 5.2.3.3).

5.3.6.1 MrBayes

Perhaps the most popular Bayesian software package for constructing phylogenetic trees is MrBayes (Huelsenbeck and Ronquist 2001; Ronquist et al. 2012).

The MrBayes package provides implementations of the most common sequence evolution models, including the JC69 and K80 models mentioned above and many others. The default prior distribution for the tree topologies is uniform, which means that all possible bifurcating tree structures are considered equally probable a priori. The default prior distribution for the branch lengths is an exponentially decaying distribution that assigns less probability to longer branches. Both these choices can be changed in a number of ways.

Computationally, Bayesian methods are almost invariably as hard or harder than the corresponding “plain” (or frequentist, or classical) statistical methods. This is also true in the case of the method applied in MrBayes. The algorithm works by generating a large sample of different hypotheses, that is, tree topologies together with the associated parameter values, by a procedure known as Markov Chain Monte Carlo (MCMC). The default sample size (number of hypotheses) is one million. The sample is generated in such a way that the different hypotheses appear in it proportional to their posterior probabilities. In other words, the most probable hypotheses appear most often, and the very improbable hypotheses hardly ever appear. However, since the sample of hypotheses is finite, an element of chance is inevitable. Moreover, the whole procedure can be repeated multiple times in order to reduce the risk of unrepresentative outcomes. The default number of repetitions is two.

The input is in the format of aligned character sequences. The output is a sample of tree hypotheses (topologies with parameter values). MrBayes includes a number of techniques for summarising the sample, including a consensus tree similar to that often applied in bootstrap analysis. A benefit of the Bayesian approach is that the uncertainty in the outcome is always expressed clearly, and thus no additional sensitivity analysis such as bootstrapping is required.

The assumptions underlying the analysis are related to the adopted sequence evolution model. In addition, the chosen prior distribution should be considered an assumption as well. The prior distribution, however, is more flexible, and it is usually recommended that prior distributions are chosen to be so vague, or “flat”, that the information in the data overrules them.

Tehrani (2013) applied MrBayes to analyse the oral tradition of the fairy tale *Little Red Riding Hood*.

5.3.6.2 BEAST

BEAST and BEAST2 are two comprehensive software packages for Bayesian phylogenetic inference (Drummond and Rambaut 2007; Bouckaert et al. 2014; Drummond and Bouckaert 2015). They include the same kind of phylogenetic tree-sampling methods as MrBayes but also a large number of other phylogenetic models and methods. Like the algorithms in MrBayes, most of the algorithms in BEAST/BEAST2 are based on MCMC sampling, and they produce a set of possible results together with estimates of their posterior probabilities.

To mention an example of the alternative analyses available in BEAST/BEAST2, the multispecies coalescent model can be used to correct for misleading phylogenetic signals in cases where the underlying process involves a population of individuals rather than a single individual. Coalescent theory, developed in the 1980s by John Kingsman and others, describes such scenarios and motivates models that are somewhat more complicated than the traditional sequence evolution models discussed above (Kingman 1982).

Another type of analysis provided by BEAST/BEAST2 is phylogeographical analysis. In phylogeographical analysis, the phylogenetic tree models are combined with geographical migration models. Such an analysis can be used, for example, to trace epidemics or the migration of populations (Lemey et al. 2009).

5.3.7 Stemmatology-specific methods

The development of methods tailored for stemmatology is not common. This is probably in part because of the success in applying phylogenetic methods proper in the computer-assisted working mode discussed at the beginning of this section. Another part of the explanation for this is the fact that the field itself is relatively young – compared, for instance, to the more mature field of phylogenetics. Moreover, by

modifying the data and the parameters of the methods, many possible deficiencies (see 5.5) of phylogenetic methods with respect to stemmatological applications can be alleviated to a degree that is often sufficient. It is reasonable, however, to expect that even better results can be achieved by designing methods for stemmatological needs from the outset.

5.3.7.1 RHM

The Roos–Heikkilä–Myllymäki (RHM) method resembles the maximum parsimony method (see above). In particular, its key idea is to minimise the amount of change along the branches of the tree. However, in contrast to parsimony, the RHM method measures the amount of change in terms of textual similarity instead of the number of different characters. The textual similarity comparison is done on a segment level rather than on a word-by-word level. The default length of a segment is ten words. To compare two segments, the RHM method applies a data compression measure (Roos, Heikkilä, and Myllymäki 2006). The higher the number of matching substrings (sequences of contiguous letters or other symbols), the higher the measured similarity.

A consequence of the segment-level compression measure is that RHM can automatically deal with changes in word order, since a segment where the order of two or more words is exchanged is still more similar (in terms of matching substrings) to the unmodified version than a segment where the words have been changed into some other words. Similarly, the compression measure assigns higher similarity scores to changes where a word is changed only slightly than to cases where a word is changed completely. A possibly problematic feature is that longer words are assigned higher importance since they contain more substrings, and changing them therefore leads to a greater decrease in the similarity score.

Since the RHM method operates directly on the text, the input is an aligned word table instead of the character table in, for example, maximum parsimony. The encoding of the words can still be adjusted by, for example, removing punctuation or word capitalisation if they are considered unimportant or even misleading from the genealogical point of view. The output of RHM is an undirected tree with no branch lengths.

The assumption underlying the RHM method is that the text evolves independently along the branches in a “compression-parsimonious” fashion. Loosely speaking, this means that changes that are smaller according to the compression measure are more likely than bigger changes. To measure the reliability of the resulting stemma, the bootstrap method can be used.

Roos and Heikkilä (2009) compared the RHM method and nine other methods on a suite of three artificial benchmark datasets where the correct stemmata are known. The results suggest that, especially for large and complex datasets, the RHM and maximum parsimony methods outperform the other methods, including neighbour-joining and least squares, but more comparisons would be needed to obtain more conclusive results.

5.3.7.2 *Leitfehler*

Philipp Roelli and Dieter Bachman (2010) proposed a method that encapsulates the principle that a stemma should be consistent with significant variants. The method automatically assigns a score between 0 and a fixed maximum (for example, 20 or 50) to each variant position based on how likely it is to be a *Leitfehler* (see 4.3.1). This score is used to weight the variants and to compute pairwise distances between each pair of witnesses. The weights are then used to construct a tree by a distance-based phylogenetic method.

The key idea in assigning the scores mentioned above is as follows. Let A denote a *locus* in the text, and A^1 and A^2 denote a pair of variant readings at *locus* A . Similarly, let B denote another *locus* with readings B^1 and B^2 . If both A and B are *Leitfehler*, it holds that, no matter what the true stemma is, as long as there is no contamination, we can only expect to observe three of the four possible combinations (A^1, B^1) , (A^1, B^2) , (A^2, B^1) , and (A^2, B^2) . For example, if we let A^1 and B^1 denote the archetypal readings, and A^2 and B^2 denote the derived readings, then it should be virtually impossible that all three combinations (A^1, B^2) , (A^2, B^1) and (A^2, B^2) are present in at least one witness each. This follows from the fact that the “mutations” A^2 and B^2 must have occurred either one after the other in the same branch or in separate branches. We can observe variants (A^2, B^1) and (A^2, B^2) but not (A^1, B^2) in case A^2 emerged first followed by B^2 , (A^1, B^2) and (A^2, B^2) but not (A^2, B^1) in case B^2 emerged first followed by A^2 , or (A^2, B^1) and (A^1, B^2) but not (A^2, B^2) in case the variants emerged in separate branches. Thus, this approach treats occurrence only once in the tradition as the defining property of *Leitfehler*.

The algorithm of Roelli and Bachman uses as its input a variant table obtained after normalisation (see 3.3.2). Any readings with more than two variants are converted into absence–presence form to obtain only two-valued readings. Each candidate *Leitfehler* is then scored by counting the number of other candidates such that the above property, namely that at most three combinations appear, holds. So, for example, the score of *locus* A is determined by checking whether this is the case for *loci* A and B , *loci* A and C , and so forth. The candidate with the highest count is assigned the maximum score (e.g. 20 or 50), and the others are given scores in proportion to their counts. Roelli (2014) has also proposed more advanced means of weighting the obtained score.

The pairwise distances between the witnesses are calculated as sums of differences weighted by the *Leitfehler* scores of the *loci*. The distance matrix is used to construct an unrooted tree using the least-squares method of Fitch and Margoliash (see 5.3.5.1 above).

5.3.7.3 The Coherence-Based Genealogical Method

Partly heuristics-based, partly statistics-based, the Coherence-Based Genealogical Method (CBGM) could be called a hybrid approach (see also 7.1.2.2). It was developed starting in the 1980s by Gerd Mink at the Institut für Neutestament-

liche Textforschung (INTF) in Münster. It was specifically developed for the editorial work on the major critical edition of the Greek New Testament, which presents a highly contaminated situation – the *Codex Sinaiticus*, for instance, has some 23,000 corrections in about eight hundred pages, or 30 per page on average (Wasserman and Gurry 2017, 21–22). In addressing this fundamental problem, CBGM allows witnesses to have multiple ancestors while also foregoing hypothetical intermediate ancestors known as hyparchetypes, which makes it easier to represent contamination. Furthermore it fundamentally relates texts rather than manuscripts. “Finally, and most importantly, the CBGM determines ancestry using a different principle. Rather than relating witnesses deductively based on shared errors, it relates them inductively using the relationship of their variants as determined by the editor” (Wasserman and Gurry 2017, 25). CBGM discerns two types of coherence. *Pre-genealogical coherence* is based on the percentage-wise agreement between two witnesses. This overall agreement is utilised to determine whether specific agreements are coincidental or not. *Genealogical coherence* is based on tracking editors’ decisions: “At each such point where a variant is either prior or posterior to another variant, the computer tracks which witnesses attest each variant and then uses this to compile the information that constitutes genealogical coherence” (Wasserman and Gurry 2017, 28). The same decision-tracking process serves to support the consistency of editors’ work. A full explanation is presented in Wasserman and Gurry (2017) and Mink (2009).

5.4 Software tools

Armin Hoenen

This section is meant to provide a rough overview of currently available tools for the creation of trees, their usage, and their dissemination. For a technical understanding of the algorithms implemented in these tools, see sections 5.2 and 5.3. Since software is part of a volatile digital ecosystem, it is subject to constant flux and change (updates, new developments, end of support or availability, operating system innovation and shift, and so on); the information provided here is almost by definition at risk of obsolescence once it is published. Readers and users should therefore always compare more recent information to what is given here. In addition, the information here is necessarily selective and the links provided in this section can thus be no more than a snapshot of what is available at the moment of writing.

5.4.1 Background

If digital stemmatology is understood as a whole domain, the physical manuscripts or fragments can be viewed as the initial input and the stemma as final output.

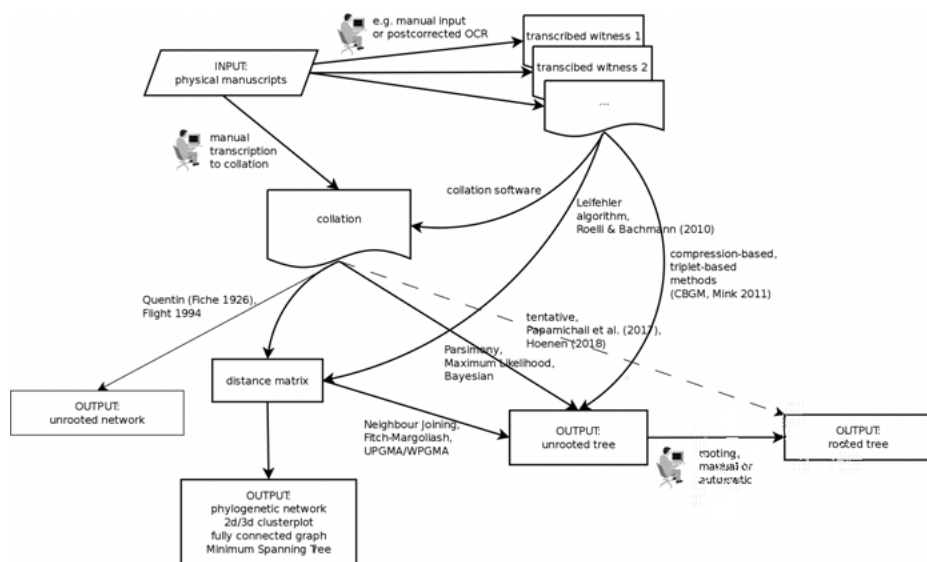


Fig. 5.4-1: Resources (nodes) and transformation processes for different ways to generate a stemma computationally (including processes that can be conducted “manually”). Terms that appear in the graphic are explained throughout the text of this section.

Between these two lie various transformations which can be achieved with the help of various tools. As can be seen in figure 5.4-1, there are many alternative ways to produce a stemma (how to arrive from ultimate input at ultimate output). Depending on which way (or technically, pipeline) one chooses, there will be different tools to use.

Tools in table 5.4-1 below are grouped by function (primarily) and then by the field they originated from (i.e. stemmatology, phylogeny, computer science, and so on). Functions proceed from collation (alignment) to tree creation. Tools may differ in aspects that apply to software in general and not to stemmatic principles alone. Are they designed for exclusively online or offline usage? Do they provide a special graphical interface? Are they accessed through a browser? Do they include well-documented functions and settings? Are there “simple” and “advanced” settings? Are they built for solving one or multiple problems (i.e. are they algorithms or suites of algorithms)? Are they a monolithic non-extendable block or are they modular (one main program with many libraries or packages)? And, finally, another important characteristic is whether they are freely available and modifiable (“free” or “open source licensed”) or not (in which case, they are proprietary software). Since some tools are much more general and it is only some libraries for them that enable their use in stemmatology (such as the packages *ape*, *phangorn*, and others for R), they may be mentioned several times as a consequence of the order chosen for the list.

The general aim of this section is to enable the reader/user to produce a tree even from only the initial input. We start with tools for collating texts, the output of which can then be processed by the genuine tree-production tools, which come in three major flavours: distance matrices, cladistic methods (parsimony), and probability-based approaches (Bayesian), as was discussed above (5.3). There are also still other ways to produce trees (see 5.3.7), for which tools are rare. In conclusion, we wish to point out that knowing a general-purpose programming language (e.g. Python, Ruby, Java) well enough to customise or implement one's own model may by many be seen as the best tool for achieving tree construction. It is, however, not the aim of this book or this section to cover programming languages and their uses. Study books and sites on these languages abound.

5.4.2 Collation

A collation is an alignment of different versions of one work (see 3.3). Manuscript text digitisation (transcription) can be conducted primarily in two ways. The first is manually, where basically any software can be used that processes text. Some scholars, however, use software, such as the Classical Text Editor (CTE), where they create a base version of the text and then reuse this as the exemplar for a new transcription, just editing the differences in order to remain in one tool, which in this case serves its purpose since CTE can lay out critical editions afterwards. The other way of transcribing is the use of manually post-corrected OCR if the source's fonts are OCR-readable – which, unfortunately, for most historical texts dating before roughly 1750 is usually not the case. There are also tools that specifically facilitate transcription by hand from images of manuscripts, such as T-Pen (t-pen.org/TPEN), Transkribus (transkribus.eu), or TextGrid (textgrid.de/en). This section attempts to cover exhaustively those tools that have been applied in stemmatological research papers. Finally, some more general, widely used tools from computer science will be listed that can be employed to support stemmatology-related tasks (as in the phylogenetic case, this is a far from exhaustive selection). Collation tools specific to stemmatology are:

- Juxta (juxtasoftware.org) is described by the website as “an open-source tool for comparing and collating multiple witnesses to a single textual work”. It is a stand-alone desktop application for input data in .txt or .xml formats.
- CollateX (collatex.net) is a multipurpose stand-alone application without a graphical interface which produces alignments of texts, offering a choice of different algorithms and output formats, including graphical output as a variant graph.

5.4.2.1 Bioinformatic and computer science tools for collation

Alignment software outside of stemmatology is widely available in bioinformatics or computer science. For instance, file difference analysers and editors can be used

to produce pairwise and sometimes multiple file alignments. The Unix command-line tool “diff” and the more text-oriented “wdiff” come natively with many Unix distributions. They identify the differences between two texts. ClustalW and ClustalX (clustal.org) are widely used tools for performing multiple sequence alignments in bioinformatics. Which of the many alignment tools from computer science and bioinformatics one prefers to use may depend on the specific tradition, the presence or absence of UTF-8 characters, and the algorithm one would prefer for alignment: whether gaps should be minimised, a weighting scheme should be possible, and so forth.

5.4.2.2 Manual collation

Of course, a collation can also be produced manually. The process then involves software, typically tabulation software such as (free and open) LibreOffice Calc or (non-free) Microsoft Excel, where texts of different witnesses can be entered side by side, each in one column or row. CTE has been used for manual collation as well. For manual vs automated collation, see section 3.3.3.

5.4.3 Distance matrix generation

A common way to produce a tree is from a distance matrix of pairwise witness distances (see 5.2.2, 5.3.3). In biology, such methods are the ones most commonly used to analyse DNA sequences, and many tools offer the possibility to produce a tree from a distance matrix as input. This is why we mention the tools appropriate to this end separately and first. Distance matrix generation requires as input data a collation and produces as output a pairwise distance matrix, that is, a table with one field for each possible pair of witnesses from the collation showing a value for that pair’s distance (see fig. 5.5-2 in 5.5 for an example). Distance itself can be calculated simply as the (relative) number of agreements or disagreements (Hamming distance; Hamming 1950), but many other distance metrics exist, some more sophisticated than others, for example Damereau–Levenshtein (Damerau 1964; see 5.2.3 above) or phonetics-based distances (Downey, Sun, and Norquest 2017). If distance measures operate on strings, they are also called string distance measures, a subclass of edit distances. Apart from Stemmaweb, there seems to be no specifically stemmatological tool allowing scholars to convert their collation to a distance matrix or to pseudo-DNA, which would be required to easily produce distance matrices or trees with bioinformatic software. Some of the following tools produce distance matrices during computation and save them somewhere, while the overt output may just be the tree. Most of the programs, however, also allow uploading a distance matrix from which one then can test different tree-generating algorithms operating on distance matrices. The most widely used tree algorithms are neighbour-joining, UPGMA/WPGMA, and Fitch–Margoliash (see 5.3.3.2–3, 5.3.5.1, 8.1).

5.4.4 Tree generation

5.4.4.1 Tools for the generation of a tree from raw or pre-processed data:

A theoretical overview

This overview discusses briefly the types of trees one can obtain from (mainstream) tools and standard post-processing procedures. As illustrated in figure 5.4-1, a tree can be generated in a number of ways: through a distance matrix, through statistical approaches such as maximum likelihood or Bayesian inference, or through cladistic approaches such as parsimony. Minimum spanning trees (5.3.3.1) can be retrieved from a pairwise distance matrix. Further ways to obtain tree structures are stepwise clustering approaches such as hierarchical clustering.

The kind of tree or, more generally, graph one produces may be of many kinds and should not be called a stemma unless it has certain properties such as an assigned root (see also 4.1). A classical stemma a priori for closed traditions is a rooted tree (rooted DAG) in graph-theoretical terms (see 4.2). Small amounts of contamination may be dealt with while maintaining the tree as the predominant visual structure, for instance in minimum hybridisation networks (see Huson and Scornavacca 2012). The most important kinds of graphs and trees that can be produced today using stemmatological, phylogenetic, and computer science software are (a) unrooted bifurcating trees with extant input data units (witnesses) at leaf positions and hypothetical ancestral nodes, and (b) unrooted multifurcating trees with extant units at internode positions but without hypothetical nodes; see figure 5.4-2.

Semstem (Roos and Zou 2011) and the approach in Hoenen (2018b) are the only stemmatological algorithms currently known to the author that produce trees which are both multifurcating and which have extant units at the internode positions. Rooted trees of this form would (apart from contamination) correspond to stemmata in philological practice. In order to turn other tree topologies (as shown in fig. 5.4-2) into formats closer to stemmata, tools may offer automatic post-processing methods. Alternatively, one can always modify the trees manually to achieve these results (rooting, conferring nodes on internode positions, collapsing bifurcations, and so on).

Unrooted trees can be made into rooted trees by applying rooting. Apart from outgroup rooting (see 5.2.1, 8.1) and midpoint rooting – which both hardly apply to

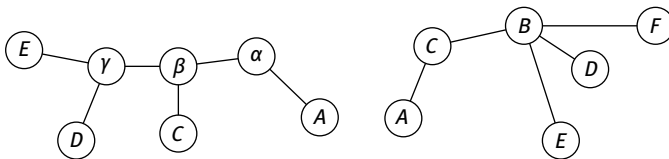


Fig. 5.4-2: Automatically producible (mainstream) tree types. (Left) unrooted bifurcating trees with extant texts (Latin letters) only at leaf positions and hypothetical nodes (Greek letters) are the output of many bioinformatic programs (although usually visually presented differently). (Right) an unrooted tree without hypothetical nodes, obtained as output of a Minimum Spanning Tree algorithm.

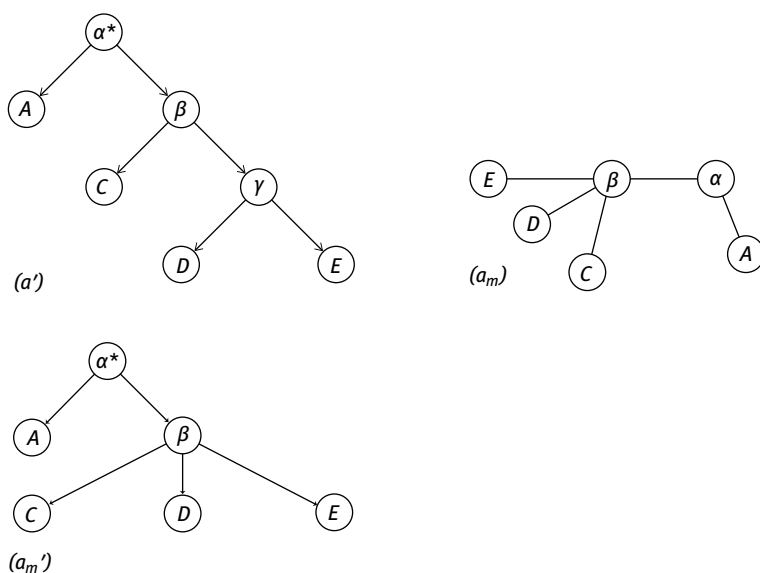


Fig. 5.4-3: Post-processed trees.

stemmatology – Marmerola et al. (2016) apply three rooting methods in the context of multimedia phylogenies (that is, phylogenies for digital media, mainly images and video; Marmerola et al. 2016, 2). One of these appoints any node in the unrooted tree as root, then counts the sum of sums of weights of all paths from the root to any other node, and finally chooses the minimum cost tree (MCT). Hoenen (2019) presents another method that attempts to detect directionality of changes. A statistical (and not strictly post-processing) method can transform bifurcating trees into trees that are multifurcating by collapsing splits below a certain level of reliability: bootstrapping consensus (see 5.2.4, 5.3.5.3; compare fig. 5.4-3). There are also approaches that turn minimum spanning trees into bifurcating trees with extant species at leaf positions (J. Yang et al. 2011).

In figure 5.4-3, a' represents is a bifurcating tree which has been rooted (root indicated by an asterisk; the tree is equivalent to the left-hand one in fig. 5.4-2); a_m shows a (hypothetical) bootstrap consensus tree for the same data: the method computed that the split at the γ node was not significant and subsequently collapsed it. Note that, owing to the lack of unifurcations in a bifurcating tree, multifurcating trees obtained from a prior bifurcating tree will not contain unifurcations. Finally, a_m' shows a rooted multifurcating tree with extant texts only at leaf positions obtained from rooting a_m .

There is no general consensus as to which kind of method for tree generation is to be considered most appropriate. Some of them (e.g. minimum spanning trees) are very different from the others, and all may have features that are not very suitable for stemmatic analysis. For instance, a phenomenon called long-branch attrac-

tion is considered problematic for cladistic and maximum likelihood-based methods (see 5.2.4.1, 8.2.5). The literature on bioinformatic research features publications that analyse which methods imply what caveats and dangers; consider Felsenstein (2004) as a starting point for further reading.

5.4.4.2 Stemmatological tools

Stemmaweb (stemmaweb.net) is a platform that provides an online graphical interface for stemma generation. It is a suite of tools that also includes a remote service (Stemweb) offering algorithms that can be used to produce unrooted trees. The input is a collation (so the program does not collate by itself) and the output an unrooted tree. Apart from an unrooted tree as output, the user gets innovative visualisations such as a variant graph and a stemma visualisation which marks variants that go against a particular stemma (if a root has been assigned) and is useful for exploring the hypotheses implied by a certain stemma. Stemweb produces trees using the RHM (Roos and Heikkilä 2009), Semstem (Roos and Zou 2011), and neighbour-joining (Saitou and Nei 1987) algorithms. The first two of these algorithms are adapted to, and in part originated from, stemmatology. A manual for the offline installation of Stemmaweb and all dependencies is available.

Stam (cosco.hiit.fi/Projects/STAM) is another project providing an interface that can be used offline and that allows the inference of stemmata using one of the stemmatologically adapted algorithms such as RHM or Semstem. Users should be aware that both Stemmaweb (including its remote Stemweb service) and Stam are very much experimental projects and that indefinite maintenance or uninterrupted service thus cannot be guaranteed.

5.4.4.3 Phylogenetic tools

Both the stemmatological tools mentioned in section 5.4.4.2 are rather recent developments compared to phylogenetic programs. Because of the pivotal task of understanding the relationships between species, the detection of evolutionary relationships is an area of much interest in biology. It is thus unsurprising that this field has attracted much attention and research. The landscape of specifically stemmatological tools is not even close to that of phylogenetics in magnitude or specificity. Phylogenetic tools were readily available and applied to stemmatology before stemmatological ones could be developed, and the results have led to a number of publications (see 5.4.7). The tools listed below can be or have been applied to stemmatological data. It must be noted, however, that careful reflection on their internal workings and on the results they produce is necessary to establish their appropriateness for the stemmatological task at hand. Both input and output will necessarily have to be adapted in order to use these tools on text data. Computer programs are constantly subject to change (updates), new releases, and obsolescence. In the same vein, documentation, dissemination through manuals and tutorials, as well as blog

posts by users experiencing and solving problems are dynamic and ephemeral. To underline this point, the first of the following tools started out as one for the computation of parsimonious trees, as the initial name “Phylogenetic Analysis Using Parsimony” (PAUP) suggests. But it has grown into an all-round tool allowing the application of various methods. Hence, a superscript asterisk has been added to its name, which signals “and other methods”. The trend for tools is to incorporate more and more functionality. Keeping track of all phylogenetic software, functionality, and methods is a Sisyphean task that can only very partially be accomplished in a handbook. To mitigate this problem, we refer here also to quite extensive online resources which try to enumerate and reference all available software packages in phylogeny and which also list their characteristics. First, however, we will list the packages that have been used in stemmatology.

- PAUP* (paup.phylosolutions.com) has been the most widely applied phylogenetic program in stemmatology. It offers an easy-to-use graphical interface with many options, algorithms, and parameterisation options. It is very well documented and available on all major operating systems. It comes with a commercial license. For details, see the website. For an overview of the functionality of PAUP*, we refer to the links listed at evolution.genetics.washington.edu/phylip/software.html.
- Phylip (Felsenstein 1993; evolution.genetics.washington.edu/phylip.html). Similar to PAUP*, this is a stand-alone program with an interface that allows the user to input, for instance, a DNA alignment and to compute trees using different algorithms. The current version of Phylip is free.
- SplitsTree (Huson 1998; splitsree.org). One of the features of this program is an effective implementation of the split decomposition algorithm invented by Bandelt and Dress (1992). This algorithm produces networks instead of trees.
- MrBayes (mbisweden.github.io/MrBayes) is a suite of programs that offers a wide range of Bayesian (probabilistic) methods for producing trees.
- Phylogeny.fr (phylogeny.fr). While the above packages are stand-alone applications, this one is a collection of online tools for tree creation that incorporates some of those already mentioned, especially Phylip. It offers a wide range of input and output formats and of tree-generating algorithms. It does not require download or installation; data is processed through the phylogeny.fr servers.
- The R libraries phangorn, ape, and RPhylip. R is a general statistical programming language which is open source and available free of charge. The packages in question are provided by users to other users and offer a wide range of programming and visualisation functions in connection with bioinformatics. They can be used to compute virtually anything within phylogeny: trees from collations, roots, and so on. Many online tutorials for R are available.
- LisBeth (Bagils et al. 2012) is a program suite which allows the computation of a tree based on three-item analysis: infosyslab.fr/?q=en/resources/software/lisbeth/download.

- molbiol-tools.ca/Phylogeny.htm is an annotated list of tools from bioinformatics. Among its entries is T-Rex, a program that allows the inclusion of contamination-like structures.
- There is an extensive list of phylogenetic software currently documenting roughly 450 different sources as well as links to other lists at evolution.genetics.washington.edu/phylip/software.html.

Lastly, a word of caution about searching for tree-generating software on the Internet: the tree is a biological metaphor, and real physical trees obviously exist as well. There are programs which simulate the growth of natural trees, and these programs are naturally also “tree-generating programs” (see e.g. the list at vterrain.org/Plants/plantsw.html).

5.4.4.4 General computer science tools

Almost all major programming languages offer phylogenetics-oriented libraries (components and extensions in the same language that may come with the language but must often be installed separately). These libraries compute graphs and allow tree generation. They often include implementations of well-known tree-generating algorithms from bioinformatics or other origins. As a cursory example, a library for Java may be mentioned:

- `jgraphT`: a Java library including algorithms to produce trees, for instance, from pairwise distance matrices.

5.4.5 Tree visualisation

Apart from tools for generating trees, there are tools that specialise in visualising trees – for instance, visualising the non-graphical output of a phylogenetic program.

- `DynStem` (github.com/ArminHoenen/dynamicStemma; Hoenen 2016) describes how to dynamically generate a stemma from Newick input and mentions new visual tree formats such as `circular tree maps`, which depict trees as circles within circles (see fig. 5.4-4).
- `FigTree` (tree.bio.ed.ac.uk/software/figtree) is a phylogenetic tool mainly for visualising trees. For instance, it offers midpoint rooting of unrooted trees.
- `Gephi` (gephi.org) is a tool primarily used for graph visualisation, including, but not limited to trees. If one has, for instance, only a list of edges, Gephi offers many designs and patterns for rendering the implied graph and allows colouring and assigning labels. Additionally, some standard graph measures such as centrality can be automatically computed, thus providing some information about, for instance, how imbalanced texts are distributed in the stemma, whether there is one very large branch, and so on.

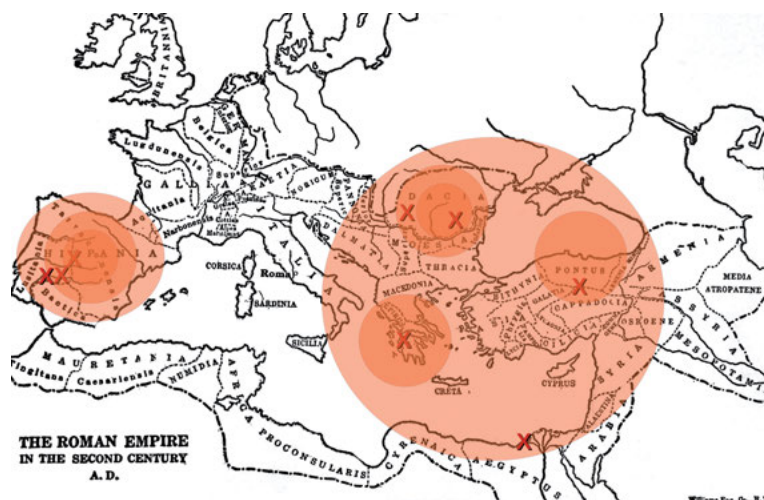


Fig. 5.4-4: A circular tree map overlaid on a geographical map. The circles stand for one witness text each, the crosses for the tentative origins of the witnesses. This is just one example of the many visualisations that can be achieved using visualisations of trees other than the usual node–edge ones. Source of geographical map: gutenberg.org/files/32624/32624-h/32624-h.html.

- Graphviz (graphviz.org), like TreeViz, provides a large number of highly customisable visualisations. Since trees are graphs, the software can be used to visualise stemmatic structures.
- igraph (igraph.org/r) is an R library with similar properties to Gephi.
- TreeDyn (treedyn.org) is stand-alone software for the post-processing of trees. One can link additional information to labels, compare different trees visually, or apply a wide range of other functions.
- TreeViz (randelshofer.ch/treeviz) is an interactive tool based on Java for the generation of visualisations of trees.

For an overview of phylogenetic tools for tree visualisation, consider Pavlopoulos et al. (2010). PhyloMap (Zhang et al. 2011) combines 2D plots with trees. Parks (2012) shows in his thesis various examples of combining trees with maps. Finally, Schulz (2011) presents a page (treevis.net) which tries to keep track of all the tree visualisation tools out there.

5.4.6 Urtext reconstruction

While at least four publications report attempts to automatically reconstruct ur-texts (i.e. hypothetical archetypal texts), namely Nassourou (2013); Hoenen (2015b); Koppel, Michaely, and Tal (2016); and Hoenen (2018a), tools allowing this are currently available only in the realm of bioinformatics, mainly using Bayesian inference.

- PAML (abacus.gene.ucl.ac.uk/software/paml.html) is an open source tool allowing, among other things, the generation of ancestral sequences along given trees (or computed trees). It includes an option to input a custom-made substitution matrix.
- BEAST/BEAST2 (beast2.org). A program for Bayesian inference which also allows the generation of archetypal sequences.

A word of caution: although inferring stemma and archetype are related problems, mathematically one stemma may correspond to many archetypal texts, and one archetypal text can be consistent with different stemmata. That is, if one solves the problem of generating a stemma (automatically), one is still faced with the task of reconstructing the archetypal text. Given one and the same stemma, imagine a bifurcation with two texts. Imagine that, at position 0 of the collation, one has variant *A* and the other variant *B*, and at position 1, one has variant *D* and the other variant *E*. The reconstructed text may have any combination (*AD*, *BE*, *AE*, *BD*), or even lost variants, but the stemma remains the same. Inferring the stemma itself and reconstructing the text (deciding on significant variation and on original variants) are deeply intertwined in the classical, manual methods. For the computer, however, either task can be executed independently. Some of the tree-generating methods take into account variant configuration, for instance parsimony, others only operate on somehow variant-neutral distances. Devising a most likely archetypal text does also not necessarily determine genealogical relationships either. Koppel, Michaely, and Tal (2016) use an expectation maximisation-based approach for urtext reconstruction where no genealogical classification or stemma is involved.

5.4.7 A small empirical survey of tools used in stemmatology

Table 5.4-1 contains a non-exhaustive list of publications that have published stemmata generated by computational tools in the last few decades.

Tab. 5.4-1: A list of some publications that have published stemmata generated by computational tools in the last few decades.

Program	Publications that use it
PAUP/PAUP*	<p>Lee (1989)</p> <p>Robinson and O'Hara (1992)</p> <p>Robinson and O'Hara (1996)</p> <p>Salemans (1996)</p> <p>Robinson (1996a)</p> <p>Salemans (2000)</p> <p>Spencer and Howe (2001)</p> <p>Spencer, Wachtel, and Howe (2002)</p>

Tab. 5.4-1 (continued)

Program	Publications that use it
	Mooney et al. (2003) Spencer, Bordalejo, Robinson, et al. (2003) Spencer, Bordalejo, Wang, et al. (2003) Macé, Baret, and Lantin (2004) Lantin, Baret, and Macé (2004) Spencer, Mooney, et al. (2004) Spencer, Davidson, et al. (2004) Yorav, Dagan, and Graur (2005) Windram et al. (2008) Phillips-Rodríguez, Howe, and Windram (2009) Heikkilä (2014) Halonen (2015) Robinson (2015)
SplitsTree	Barbrook et al. (1998) Mooney et al. (2003) Stolz (2003) Spencer, Mooney, et al. (2004) Eagleton and Spencer (2006) Windram et al. (2008) Heikkilä (2014) Halonen (2015)
Phylip	Macé, Schmidt, and Weiler (2001) Woerther and Khonsari (2003) Roelli and Bachmann (2010) Heikkilä (2014) Roelli (2014)
others	Roos and Heikkilä (2009), own Le Pouliquen (2010), own Roos and Zou (2011), own Hoenen (2015b), PAML Papamichail et al. (2017), own Hoenen (2018a), own Lee (1989), MacClade

As one can see, PAUP/PAUP* has been used overwhelmingly most often. Apart from this, Phylip and SplitsTree are also commonly used, other programs only occasionally. With many other programs available, this may change in the future.

5.5 Criticisms of digital methods

Jean-Baptiste Guillaumin

Over the past few decades, digital methods of stemmatology have given birth to a new field of research at the interface between philology, biology, and computer science; a history of this approach can be found with a full bibliography in section 5.1 and in Trovato 2017 (chap. 3.2: “A Brief History of Computer-Assisted Stemmatics”). These methods, at first strictly based on bioinformatic algorithms (distance matrix-based methods, parsimony methods, maximum likelihood, and Bayesian inference), have been specifically adapted for stemmatology in some recent studies, for example with RHM or Semstem algorithms (see 5.2–3); more rarely, digital methods for stemmatology have also been developed within the field, without any reference to bioinformatics, for example by Camps and Cafiero (2014). All these approaches have also been tested on artificial traditions (Baret, Macé, and Robinson 2006; Roos and Heikkilä 2009; Roos and Zou 2011), and several software tools can now be used by philologists (see 5.4).

Generally speaking, these methods can be useful guides for philologists when representing a rich tradition, for they make it easy to visualise the clearest cases of kinship, even if they do not aim (nor claim) to produce proper stemmata taking into account all the historical features of a complex textual tradition. However, at the moment none of them is able to produce a proper stemma taking into account all the subtleties of such a tradition. Since the onset of these digital methods, philologists specialised in various fields using different linguistic corpora have highlighted some limits of these approaches, whether they have used them or not (see e.g. Hanna 2000; Cartlidge 2001; Love 2004; Bland 2005; Reeve 2011b, esp. 387–399); recently, Alexanderson (2018) radically criticised the application of phylogenetic methods to textual history. Moreover, other general studies on computer-assisted stemmatics have voiced some criticisms (see Robins 2007; Trovato 2017, chap. 4), and given rise to replies from specialists in this field (C. J. Howe, Connolly, and Windram 2012; Bordalejo 2016; Macé 2019). Some users of these methods have also taken advantage of their experience to highlight some unresolved issues (Roelli and Bachmann 2010, 329–331; Roelli 2014; Heikkilä 2014). The goal of this section is to summarise the most common of such criticisms, not in order to deny how useful the digital approach can be, but rather to assess for what issues a traditional philological approach cannot be relinquished at present. In such a topic requiring a high level of interdisciplinarity, it would be very difficult for a single person to fully understand all the approaches that have been developed and to evaluate their philological efficiency with real textual traditions: since the author of this contribution is not a computer scientist or a biologist, but a philologist who once tried a few of these methods for his own purposes, he does not claim to have a comprehensive view of all the algorithms presented above. Most of this section will therefore concern dis-

tance matrix-based methods, but when a specific kind of criticism is also valid for other methods (which is very often the case), this will be stressed.

5.5.1 Criticism of the phylogenetic paradigm and possible responses

A first type of criticism that can sometimes be found deals with the very possibility of an analogy between textual traditions and phylogenetics (Alexanderson 2018, 387–396). It is mainly based on the fact that the texts were copied by human hands which might induce changes either because of the copyist's negligence or, on the contrary, because of his clever interventions such as spontaneous corrections or search for better readings through contamination. From this point of view, the analogy with the natural evolution of species seems difficult to maintain.

Several answers to this kind of criticism are nevertheless possible. First, as has sometimes been emphasised, some ground for comparison can still be found between such interventions and the field of phylogenetics: contamination can be compared to recombination, which is also a difficult issue for phylogenetics; just like textual mistakes, some mutations in biology are reversible (see C. J. Howe, Conolly, and Windram 2012, 57–60). Caroline Macé, in her response to Bengt Alexanderson (Macé 2019), draws a parallel between “negligence” and “intention” on the one hand, and “hazard” and “necessity” on the other, following the terminology of the biologist Jacques Monod. More generally speaking, even if differences between the two fields are undeniable, drawing an analogy does not necessarily mean transposing exactly from the one to the other: using a metaphor sometimes enables a better understanding of complex issues. Besides, the notions of a stemma, a family of manuscripts, kinship between them, and so on are themselves metaphorical (see 2.2.2). It seems, then, legitimate to rely on this kind of analogy to explore a methodological convergence.

Of course, this methodological convergence does not mean that one can apply to texts, without critical thinking, the software developed to compare DNA sequences. It deals rather with a similar representation of the process of “descent with modification”, as Darwin put it (as Macé 2019 recalls); it thus opens up the possibility of the same formal treatment. Intuitively speaking, the different stages of the copying process make the distance between witnesses grow: in the same way, the distance between two species appears as a result of changes through evolution, transmitted from common ancestors. According to this model (used by all the methods relying on distance matrices), the distance between two witnesses is inversely proportional to their degree of kinship. If one of the witnesses appears as very “distant” from its direct model due to the negligence or interventionism of the copyist, it will theoretically appear as more distant from other copies of the same model that were written by a meticulous copyist who conserved a text very close to the original. The intuitive notion of distance then leads to a mathematical problem, which is to find the tree structure able to represent in the most appropriate way the set of distances between the pairs of witnesses. This problem can be solved

in a satisfactory way by the neighbour-joining algorithm (Saitou and Nei 1987, improved by Studier and Keppler 1988), as has been mathematically proved (see e.g. Mihaescu, Levy, and Pachter 2006). Thus, this approach aims less at importing some specifically phylogenetic methods into philology than at using a similar mathematical analysis in order to solve an analogous problem.

From our point of view, the analogy between phylogenetics and stemmatics is valid, but this kind of criticism encourages us to keep in mind this fundamental principle: when using these methods, the philologist must know exactly what he is doing and be able to check the different stages of the software process, which should never remain a “black box”. He also needs to interpret the result, which is never exactly a stemma, as we shall see, but generally appears as an unrooted tree-like structure.

5.5.2 A brief invented example as an illustration

At this point, rather than examining abstract variant readings (“a”, “b”, “c”, “d”, and so on), we invent a very simple and brief artificial tradition in order to illustrate the different issues: all the witnesses quoted below will thus be fictitious, although their text may remind the reader of an allegorical *ekphrasis* by Martianus Capella (*De nuptiis Philologiae et Mercurii* 1.11), whose symbolism (trees, numbers, and harmony) is not inappropriate in the present context. Of course, the aim of this discussion is not to test the validity of the processes, which would require more extensive traditions; as Roelli and Bachmann (2010, 314) say, “the longer the excerpted text, the more reliable the result is going to be” (indeed, artificial text traditions already used for this purpose deal with hundreds or thousands of words; see the references in the introduction to 5.5). Our purpose is only to show practically how they work in order to comment on the graphs that are produced. As has been said above, most of the practical treatments will use distance-based methods. The constraints on section length pre-empt testing our artificial tradition with all the other existing methods. Of course, it would be a good exercise for the reader to undertake this kind of experimentation in order to obtain another illustration of the criticisms that have been developed.

Let us, then, consider nine (or ten) words in ten witnesses whose relationships are all assumed to be known (any resemblance to existing manuscripts being purely coincidental):

- A (ninth century): “Eminentiora prolixarum arborum culmina perindeque distenta acuto sonitu resultabant.”
- B (eleventh century, copied from A): “Eminentiora prolixarum abietum cacumina perindeque distantia acuto sonitu resultabant.”
- C (tenth century, copied from A): “Altiora prolixarum arborum culmina perindeque distenta acuto sonitu resonabant.”
- D (eleventh century, copied from C): “Altiora promissarum arborum culmina perindeque distenta acutissimo sonitu resonabant.”

- *E* (twelfth century, copied from *C*): “Altiora prolixarum arborum culmina perindeque distenta acuto sono resonabant.”
- *F* (thirteenth century, copied from *C*): “Altiora prolixarum arborum fulmina perindeque et distenta acuto tinnitu resonabant.”
- *G* (fourteenth century, copied from *E*): “Altiora arborum culmina perindeque discreta acuto sono resonabant.”
- *H* (fifteenth century, copied from *E*): “Altiora prolixarum arborum culmina proptereaue distenta acuto sono resonabant.”
- *I* (twelfth century, contaminated from *B* and *D*): “Eminentiora promissarum abietum culmina perindeque distenta acutissimo sonitu resultabant.”
- *J* (twelfth century, copied from *B*): “Eminentiora prolixarum abietum cacumina per insignem distantiam acuto sonitu resultabant.”

Of course, the rate and the nature of the modifications in this brief text are particularly improbable in the real world, in which there are few cases of substantive textual innovation; improbable is also the fact that all witnesses are extant. However, this simplistic example allows a practical description of some methods, and some modifications to it (e.g. the suppression of some witnesses) will be tested below. According to the list above, the correct stemma is the one in figure 5.5-1.

Century

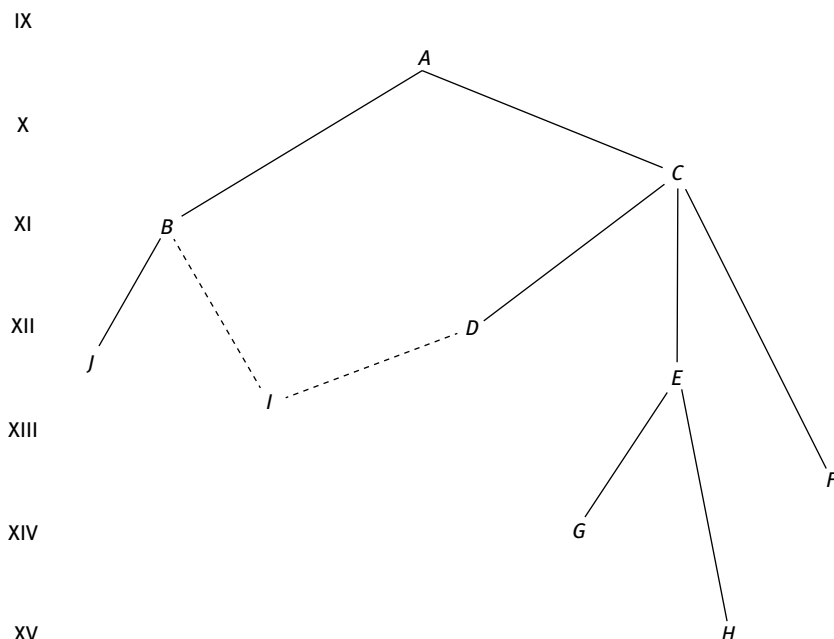


Fig. 5.5-1: Correct stemma for the artificial tradition.

5.5.3 How to calculate distances

Among all the digital methods, the most common ones are based on the calculation of a distance matrix, that is, a table which contains all the distances between each pair of items under consideration. It is then necessary to define as precisely as possible what is meant when one speaks about “distance” between two texts (see esp. Spencer and Howe 2001). Intuitively, it is possible to define distance as the number of modifications necessary to arrive from one to the other. But the proper counting of this distance can take various forms. Roughly speaking, a distance between two texts can be established by counting the number of characters or words that differ (an omission or addition being counted as one difference). With the example above, this method would yield the first matrix in figure 5.5-2 for a count based on the number of words differing between two texts, and the second matrix for a count based on the number of different characters (of course, since the distance calculation is commutative, each of these matrices is symmetrical and could be presented in a simpler way).

	A	B	C	D	E	F	G	H	I	J
A	0	3	2	4	3	5	5	4	3	5
B	3	0	5	7	6	7	7	7	4	3
C	2	5	0	2	1	3	3	2	5	6
D	4	7	2	0	3	5	4	4	3	8
E	3	6	1	3	0	3	2	1	6	7
F	5	7	3	5	3	0	5	4	8	8
G	5	7	3	4	2	5	0	3	7	7
H	4	7	2	4	1	4	3	0	7	6
I	3	4	5	3	6	8	7	7	0	6
J	5	3	6	8	7	8	7	6	0	0

	A	B	C	D	E	F	G	H	I	J
A	0	9	9	17	12	15	25	19	12	16
B	9	0	18	26	21	24	32	28	13	7
C	9	18	0	8	3	6	16	10	21	25
D	17	26	8	0	11	14	22	18	13	33
E	12	21	3	11	0	9	13	7	24	28
F	15	24	6	14	9	0	22	16	27	31
G	25	32	16	22	13	22	0	20	34	39
H	19	28	10	18	7	16	20	0	31	33
I	12	13	21	13	24	27	34	31	0	20
J	16	7	25	33	28	31	39	33	20	0

Fig. 5.5-2: Distance matrices for the example, based on different words (left) and characters (right).

Although there are some efficient algorithms for calculating this kind of edit distance (e.g. Levenshtein’s algorithm or Unix’s “diff”), both word- and character-based approaches present some theoretical inconveniences. In particular, one can intuitively see that not all the substitutions should have the same weight: graphical modifications (e.g. “i” instead of “y”, “accidere” instead of “adcidere”, and so on), or even substitutions of similar words (like “experimentum” instead of “experientia”, to take an example quoted by Trovato 2017, 194), should not get a score as great as the substitution of a completely different word (“exemplum” instead of “experimentum”; on substantive and accidental variation, see 4.1.5). A count based on characters often allows us to introduce variation between these different cases, but only in an approximate and somewhat unpredictable manner (“experimentum”/“exemplum” would be counted as 6, “experimentum”/“periculum” as 6, and “ex-

perimentum”/“periclitatio” as 9). And what about mistakes caused by erroneous word breaks (like “experimentum”/“experti mentium”), which will be measured very differently depending on whether one chooses a word-based distance or a character-based one? Moreover, and more problematically, this kind of mechanical measure does not allow us to take into account the possible syntactic or semantic reasons for a replacement (modification affecting several consecutive words, replacement of a word with a synonym, or another kind of polygenetic modification).

Despite all these theoretical reproaches, which are legitimate, the naive measurement of character-based distance is often enough to give a good idea of the kinship between witnesses. For our illustrative purposes, we will simply use here the raw character-based distance matrix calculated above. With the neighbour-joining algorithm, for example, we get the following graph description (to make the presentation clearer, numbers are rounded, if necessary, to three decimal places):

(((((A:0.917,((B:0.125,J:6.875):6.143,I:6.857):3.083):7.05,D:5.95):1.708,
(C:0,F:6):0.292):2.578,G:12.609):0.39,E:0.016,H:6.984).

This translates to the graphical representation in figure 5.5-3.

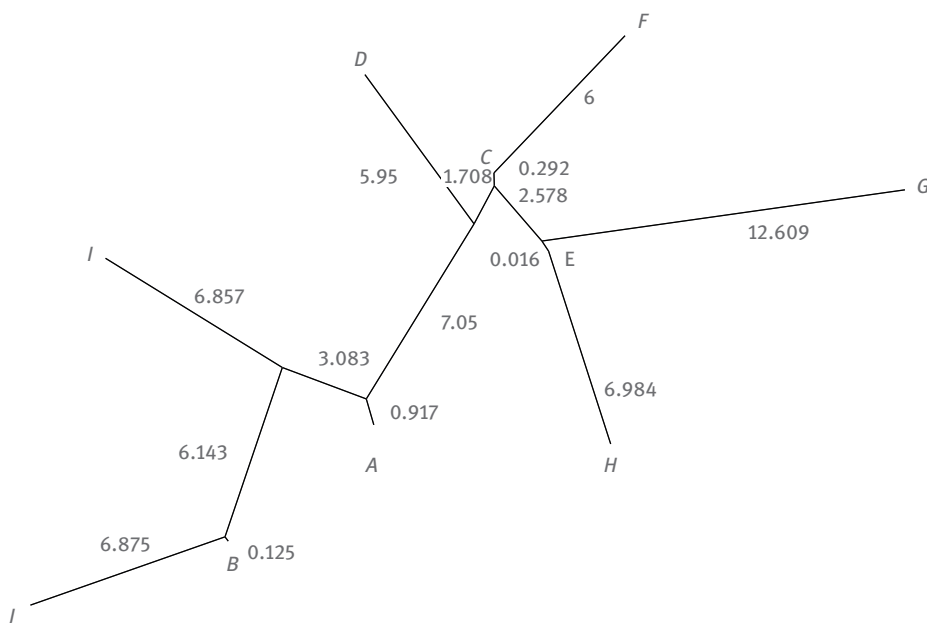


Fig. 5.5-3: Neighbour-joining graph from the distance matrix. The neighbour-joining calculation was done with purpose-made software written in Ocaml (which takes transcription text files as input, establishes a Levenshtein distance between them, and applies neighbour-joining); the graph was drawn with the “drawgram” software (part of Phylip package), and adapted for this paper; numbers were added manually to explain the link with the graph description above.

5.5.4 Noise and weighting of the readings

A comparison of figure 5.5-3 with the “real” stemma in figure 5.5-1 shows the general validity of this process, even using such a naive calculation of distances. However, beyond the points of criticism mentioned already, a fundamental question remains: is it legitimate to take all the variant readings into account without any hierarchy? This question is linked to the discussion of *Leitfehler* in (neo-)Lachmannian theory (see 2.2.5). Indeed, when one measures syntactic modifications or banal variant readings (which can be polygenetic) in the same way as mistakes introduced by a copyist at a precise moment in the transmission, “noise” risks interfering with the result, rendering the clustering less clear and less efficient.

In order to correct this problem, Roelli and Bachmann (2010, 317–318), after having chosen a word-based distance, propose an automated method to decrease the score of syntactic variants (with a parameter p between 0 and 1 applied on k consecutive edits) and weight the significant mistakes (i.e. the *Leitfehler*) by testing, for every pair of readings A and B , the distribution over the entire corpus between (A, B) , $(A, \text{not } B)$, $(\text{not } A, B)$, and $(\text{not } A, \text{not } B)$ and picking out the variants for which one of these four categories does not occur: once found, these variants are weighted with a coefficient (see 5.3.7.2). In our artificial tradition, this would be the case, for example, for (“*eminentiora*”, “*arborum*”) or (“*culmina*”, “*resultabant*”), but not for (“*eminentiora*”, “*prolixarum*”) due to the contamination of *I* rather than polygenesis. Roelli (2014) proposed improvements on this method. The idea is to use several stages to improve the appearance of the tree: although the first is automated, the following ones require from the philologist the choice of hand-picked “good *Leitfehler*”. The result seems very accurate, but one could object that this method requires a philological a priori intervention which aims not to interpret the result but rather to influence the process. Moreover, one could argue that the recurrence of banal variants may also give an idea of some kinship relations if they occur frequently at the same place in several manuscripts: if they do not, they make noise increase, apparently without a pernicious effect on the general structure. Although this point of view differs from the Lachmannian theory, taking into account all the variants, even the most trivial ones, may be justifiable; see Spencer, Davidson, et al. (2004), and Andrews and Macé (2013, 518): “even the most trivial changes, taken in aggregate, have some text-genealogical significance that should not be discounted”. But, generally speaking, the discussion about weighting variants for a more accurate result remains open, as it is in the various methods used in phylogenetics.

5.5.5 Orientation and rooting of the tree

One of the recurrent points of criticism is based on the impossibility of rooting and orienting the tree with most of the digital methods. Indeed, the UPGMA method,

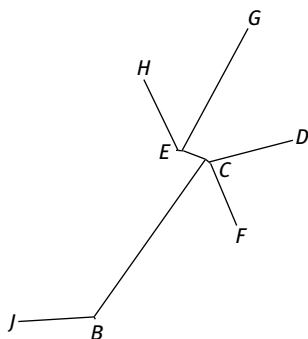


Fig. 5.5-4: The same plot without the archetype *A* and the contaminated manuscript *I*.

which produces a rooted and directed tree structure (see 5.3.3), is valid only for a constant evolutionary speed (the molecular clock in biology), but this case almost never occurs in stemmatics. As far as I know, other digital methods give neither a root nor an orientation, but an unrooted graph which the philologist has to interpret with his own methods (in particular, his knowledge of the historical background of each witness) to find the place of the root, that is, of the archetype.

One can say, using an image, that the result received from most of the digital methods looks like an articulated puppet which the philologist is to animate: an algorithm is thus successful if it gives the correct structure of this puppet (i.e. the place of the articulations for the different limbs), no matter how the philologist then decides to make it walk. Nevertheless, this limitation is not a really problematic issue; it is even useful, since the philologist himself keeps the responsibility of introducing into the graph the historical dimension of the studied textual tradition. In our example (fig. 5.5-3), the philologist should posit the archetype at *A* (or close by) because of the distribution of the variants considered as the best from a philological point of view (e.g. “*eminentiora*”, “*arborum*”, “*resultabant*”) and because of its date. If *A* is unavailable (as is almost always the case for an archetype), and if we do not take into account the manuscript *I* because of the suspected contamination (which can change the topology of the tree; see 5.5.8), an analysis of the following graph, completed with a study of the distribution of variants, should correctly put the archetype somewhere on the segment between *B* and *C* (fig. 5.5-4).

In biology, one can root a graph produced with a phylogenetic method by introducing artificially into the calculation a remote species known to belong outside the studied group (outgroup rooting; see 5.2.1). In philology, this is practically impossible insofar as, by definition, the entire available tradition has to be taken into account in the stemma (the only comparable case would theoretically be an ancient rewriting, interpolation, or translation prior to the archetype, but this kind of example is uncommon and difficult to harmonise with the distance calculation; for an example, see 4.5.2). In the future, additional methods might be developed to “polarise” variant readings, for example by determining, for each of them, whether it is

likely to be derived from another; for this purpose, a “categorisation system” could be useful (see Andrews and Macé 2013).

5.5.6 Prevalence of bifurcating trees

Another point of criticism, linked to a traditional discussion in stemmatics (see 5.1.2.1), deals with the prevalence of the bifurcating structure (i.e. structures in which each interior node has exactly three neighbours). Indeed, most of the methods presented above lead to bifurcating trees. This can sometimes be used on purpose in order to simplify the model: for example, Roos and Heikkilä (2009, 432), write that “for simplicity, and following the common practice in phylogenetics where it is perhaps better justified, we restrict the stemma to a bifurcating tree” (in their presentation of the RHM method). In other cases, this characteristic is the result of the algorithm used: thus, neighbour-joining most of the time produces a bifurcating tree because it groups taxa in pairs at each iteration – still, a multifurcating structure is theoretically not strictly impossible, since the calculation of distance between a group and a node can take a zero value: for example, with a tradition “a a a” (A), “a b b b” (B), “a b c c” (C), “a b d d” (D), and the distance system $AB = 3$, $AC = 3$, $AD = 3$, $BC = 2$, $BD = 2$, $CD = 2$, one would get the tree $((A:2,B:1):0,C:1,D:1)$, corresponding to a trifurcation (fig. 5.5-5).

More precisely, it happens frequently that a distance between two nodes in a neighbour-joining tree appears very short: in this case, when interpreting the graph, the philologist can decide to remove this distance and to take only a single node into consideration instead of both. In the following graph (fig. 5.5-6), for example, the witnesses C and E have been removed from our sample to get a simpler structure without internal nodes (on this question, see 5.5.7) and to verify that the configuration is not fundamentally modified by such an absence (as a response to the possible objection about the unlikelihood of such a complete tradition; see 5.5.2). One could thus legitimately decide to link D and F to a unique common node; in this

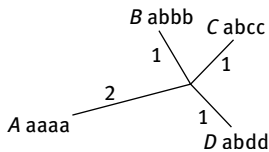


Fig. 5.5-5: A simple example of a trifurcation with neighbour-joining.

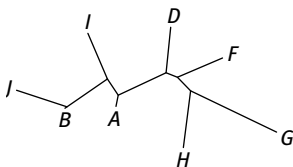


Fig. 5.5-6: The above example without the inner nodes C and E.

case, there would be a trifurcation from this common node (to *D*, *F*, and the common ancestor of *H* and *G*, namely *E*, not taken into account here).

Trifurcation is then an available option when two nodes appear very close to each other, but, as we have already pointed out previously for rooting, only the philologist has the competence to validate this kind of simplification.

5.5.7 Witnesses as internal nodes

Among the points of criticism dealing with the theoretical structure of the tree and the differences between phylogenetics and stemmatology, one can also mention the possibility that a witness appears as an internal node if it is proved to be the ancestor of one or several other(s) which are located as leaves on the tree. Indeed, this is even quite a frequent case, and it is fundamental to depict it in a stemma because it is the basis for the *eliminatio codicum descriptorum* (see 2.2.5). In phylogenetics, on the contrary, this case should not happen: the examined species are contemporaneous, and the internal nodes represent missing common ancestors. Most of the phylogenetic algorithms therefore do not offer the possibility of having a taxon as an internal node, except through manual intervention. With the distance matrix-based algorithms, it is very unusual (but not absolutely impossible, especially for a small set of data) to obtain internal nodes corresponding to witnesses still available: in the tree above (5.5.3), such a case can be observed with *C*, which is put at a zero distance from its common node with *F*. In the same tree, several internal distances are also close to zero – a far more common situation which would intuitively lead the philologist to assume a common node (observe the places of *E*, *C*, *B*). Even for a somewhat greater distance, for example between *C* and the origin of the branch of *D*, the philologist should assume a unique junction on *C* and seek to verify it with traditional methods (e.g. by picking out some *Leitfehler* and analysing their distribution): the distortion is here linked to the effects of contamination, as we shall see (5.5.8). Still in the tree above, the same hypothesis should be assumed for *A* as well.

But the fact remains that the standard bifurcating structure and the unlikelihood of getting witnesses exactly on internal nodes appear as obvious limitations for the distance-based methods, and more generally for all the bioinformatic methods, since they are a requirement of phylogenetics. This objection has been taken into account in some recent new methods specifically made for stemmatology, such as Semstem (Roos and Zou 2011). Indeed, this approach is based on the structural expectation maximisation (EM) algorithm used for phylogenetic trees (Friedman 1997; Friedman et al. 2002), but, whereas the algorithm, in phylogenetics, enables the detection and removal of non-bifurcating structures or observed interior nodes, its stemmatological use aims to confirm such features, which are quite frequent in textual criticism.

5.5.8 “Ist gegen Kontamination immer noch kein Kraut gewachsen?”

In the traditional conception of stemmatology, contamination is a serious difficulty. According to the famous adage of Paul Maas: “Gegen die Kontamination ist kein Kraut gewachsen” (1957, 31) [No specific has yet been discovered against contamination] (trans. Flower 1958, 49; on this point, see 2.2.7). This difficulty occurs also in the digital methods: for example, the tree-like approach does not allow the detection of conflation, that is, the use of two (or possibly more) ancestors in copying a unique new text. More problematically, whatever method is used, taking into account a contaminated text produces a distorting effect on the entire tree topology. Indeed, if we consider again the distance-based methods, we can see intuitively that a contaminated exemplar has quite a reduced distance from both its ancestors, even if these ancestors are distant from each other and close to other witnesses of their own families; due to the complex metric of the distance set, this distortion is even likely to produce a kind of artificial attraction between these ancestors. This is the case for *I* in the example above: since it takes more elements from *B* than from *D*, neighbour-joining introduces a common node between *I* and *B*; but, since *I* is more closely related to *D* (13) than to its ancestor *C* (21), *D*’s branching is displaced, as can be shown with a comparison between the two graphs in figure 5.5-7 (without *I* in the first, with *I* in the second).

In the case of heavily contaminated traditions, such an approach produces a massive attraction towards a point near the graph’s centre, which is absolutely not the place of the archetype (one should always keep in mind that neighbour-joining produces an unrooted tree). For example, in figure 5.5-8, a graph is plotted for a relatively brief passage (9.906–908; 245 words, containing both prose and verse) in the first hand of some manuscripts of Martianus Capella (Guillaumin 2008, 246–255), using a naive distance computation based on characters (see 5.5.3) and neighbour-joining with a dedicated piece of software (the branches of *H* and *C* have been shortened here for presentation purposes).

As one can see, there is a kind of convergence towards a point that appears as a sort of centre of gravity of contaminated witnesses; but, according to philological criteria, the place of the archetype should instead be close to the ancestor node of *A*, *H*, *R*, *B*, *D*. Very little can be said about the kinship of the branchings near the centre.

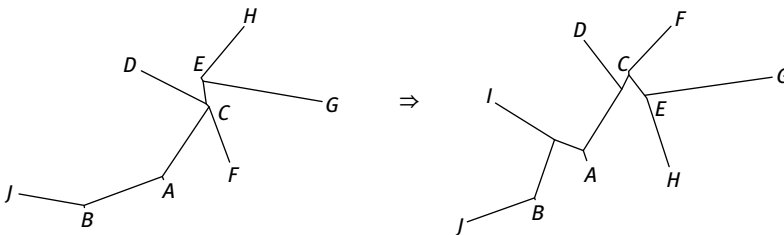


Fig. 5.5-7: The effect of the contaminated manuscript *I* on the entire tree.

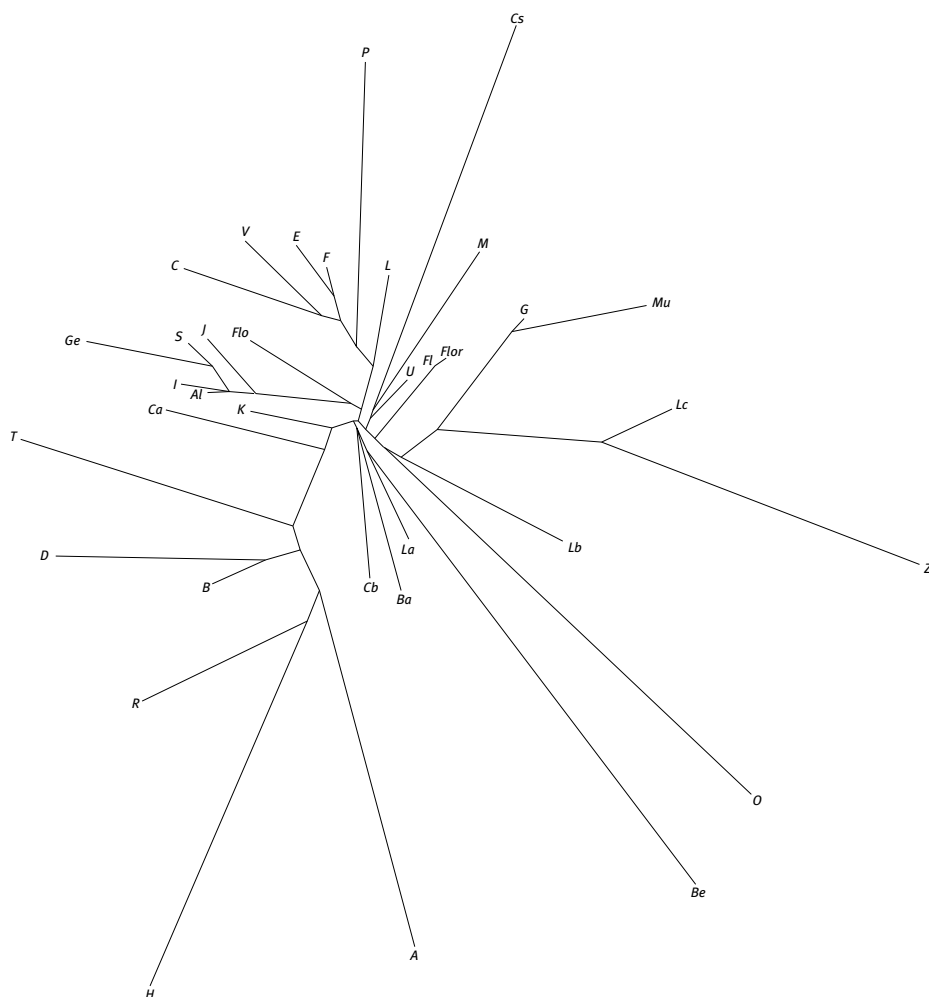


Fig. 5.5-8: Effect of contamination on a plot for Martianus Capella. The branches leading to *H* and *C* have been shortened.

For the shape of the stemma proposed for the *De nuptiis Philologiae et Mercurii* (quite unusual and probably questionable in some points), see figure 4.1-4 above.

However, there are some methods for detecting contamination: it is of course possible to test the manuscripts at different passages and look at the variations. If the copyist used sometimes one model, sometimes another, the tree topology will change from one passage to the other. Two objections are nevertheless possible: first, the delimitation of the tested sections is necessarily random, and generally unlikely to coincide with the model switch; on this point, see C. J. Howe, Connolly, and Windram (2012, 58), quoting Cartlidge (2001, 145). This method can be useful, *a contrario*, to prove the lack of contamination if the structure always remains the

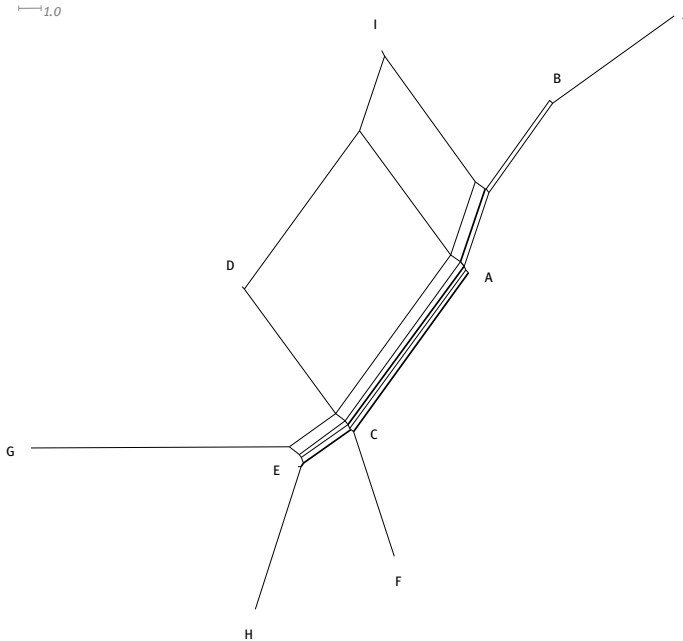


Fig. 5.5-9: Network graph edited with SplitsTree4 software, using the distance matrix presented in section 5.5.3 above.

same, no matter how many random tests are done. Second, and more fundamentally, contamination does not necessarily consist of a localisable model-switching: a copyist may have worked with two (or more) exemplars at the same time (this is the case for our artificial witness *I*). So, another solution is to use a network approach instead of a merely linear tree structure. It is possible to do this with the Neighbor-Net algorithm (D. Bryant and Moulton 2004), which has been tested in the field of stemmatology by Spencer, Davidson, et al. (2004). This algorithm takes a distance matrix as input and resorts to the same principle as neighbour-joining, but constructs a network rather than a tree, by agglomerating pairs of pairs which share one node in common. Topological irregularities thus appear like boxes, and, even if the interpretation of the graph tends to be more complex, it is then possible to detect phenomena such as contamination. In our previous example, the relation between *I* and *D* (due to contamination), complementary to the already detected proximity of *I* to *B* (quantitatively used a bit more than *D* by the imaginary copyist), can be shown as in figure 5.5-9.

However, in a strongly contaminated tradition, reading such a graph is not easy and may in the end not be very useful, except to encourage caution with regard to the interpretation of the relationships around the centre of the graph. For example, figure 5.5-10 displays the graph for the passage of Martianus mentioned above (based on the same distance matrix).

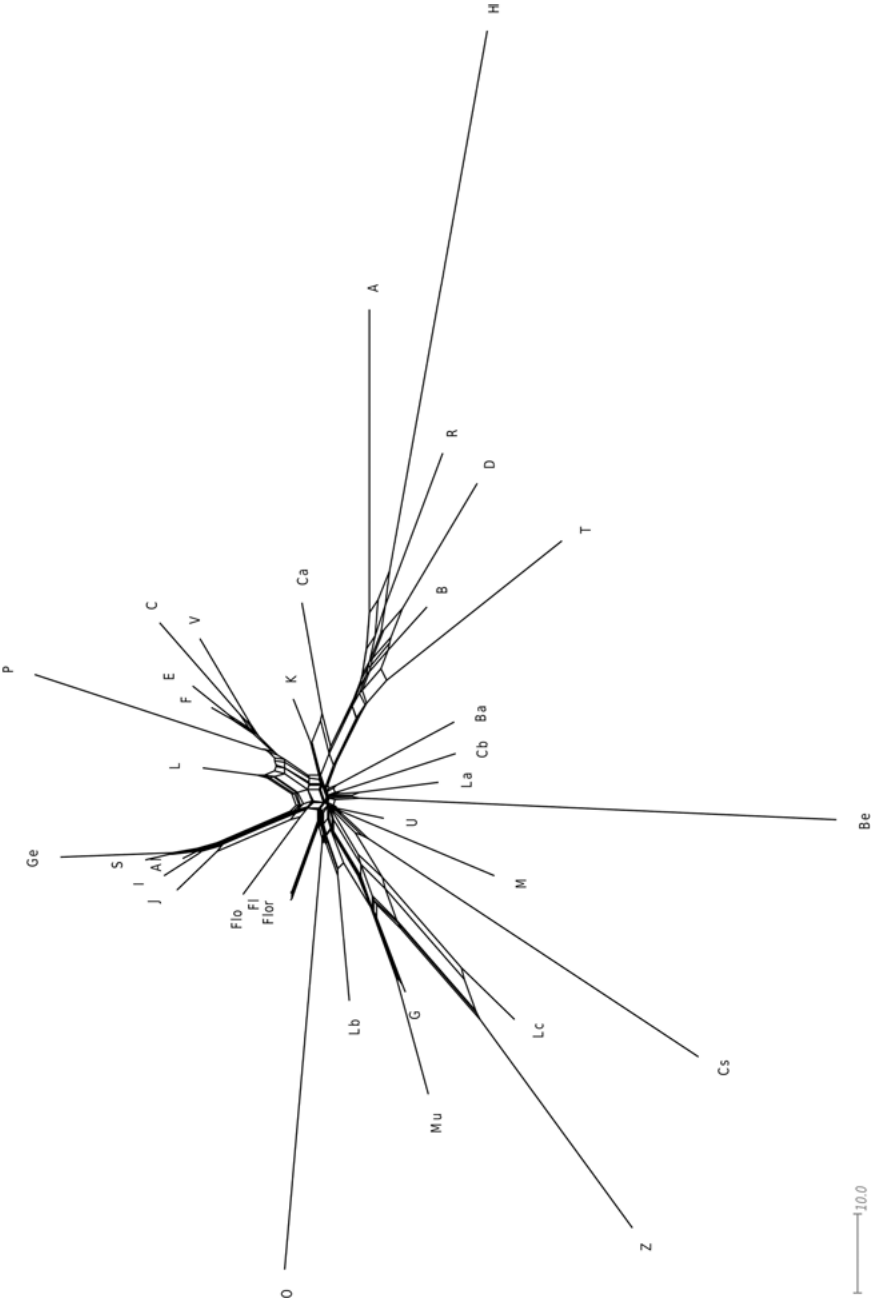


Fig. 5.5-10: Network graph for the passage from Martianus. The branch leading to H has been shortened.

5.5.9 Graphical habits, material issues and historical evolution of the manuscripts

Until now, we have considered manuscripts as if they were mere texts: this point of view is necessary for every automated approach, but it is of course simplistic, and the philologist who uses a digital method should always keep in mind that a manuscript is a material object liable to change over time.

This point concerns, for example, graphical habits, details of page layout, or even punctuation marks, all of which are likely to change over time and in different milieus: although these elements sometimes provide information that can be useful for a stemmatic approach (if they recur in the same places in several manuscripts), it seems difficult to include them in the available digital methods, for they are not real variant readings and can occur independently during the copying of a text. With the distance-based methods, it seems possible to encode this kind of detail by counting a small distance (e.g. if a manuscript has a capital letter and another a lower-case one, if one has a punctuation mark and another nothing, or if one has an abbreviation and another the full form). But the value of that distance should remain a lot smaller than that corresponding to a real variant reading, and the risk is that the noise will increase (on this point, see the discussion in 5.5.4). In most cases, due to the necessarily weak effect of these details, such a refinement would probably not change the result greatly.

However, an evolution of the material appearance of a manuscript is possible: some folios may have been damaged or even lost, and some terms may become unreadable when the parchment is worn out or scraped. In such cases, it is necessary to find a way to encode “missing data” which was not previously taken into account in the calculation, as is indeed possible with many kinds of bioinformatic software, whatever method is used. From a theoretical point of view, a first solution is to leave the *locus* out of the calculation for each manuscript, but this implies losing some information about the text of the readable manuscripts for the passage in question, which can be problematic if it contains an important variant reading. Another solution, with the distance-based methods, would be to consider that an unreadable word, unlike an omission, adds no distance, or a very small one, compared with all the readings found at the same place in other manuscripts, even if they are themselves different from one another (but this approach is also problematic if massively used, because it may contradict the “triangular inequality” that a metric by definition requires).

Moreover, the text itself will sometimes have evolved over the centuries: for example, it may have been corrected or completed with interlinear variants or glosses. Different copies of the same witness can thus be quite different depending on their dates. It is clear that this kind of textual history hardly conforms to a tree-like representation because of the cyclic graphs necessary to describe it (see Andrews and Macé 2013, 509–511). Furthermore, when a witness carries some corrections or

variants, its offspring are necessarily characterised by a form of contamination that is very difficult to model, even with a traditional stemmatic approach, since this kind of paratext often has a tradition of its own: in this case, the very possibility of drawing a stemma can become doubtful. Such copying phenomena occur, for example, in the above-mentioned manuscript tradition of Martianus Capella, so that the neighbour-joining graph discussed above (5.5.8), based only on the text of the first hand, does not show all the complexity of this contaminated circulation: to do that, one could try to introduce as an independent witness the second hand (or even the third, fourth, and so on, where it is possible to distinguish them), but the multiplication of such contaminated texts would make the graph less readable. Moreover, when a manuscript has interlinear corrections or variants, it is hard to treat it as a linear text as if there were a unique way of reading and copying it: as in the case of contamination (see 5.5.8), phylogenetic methods do not work very well for this purpose. In such a case, one could try to introduce an artificial (and quite monstrous) witness containing a concatenation of all the parallel variants in order to treat it in a linear way.

In our artificial tradition, let us assume that *B* was corrected and glossed this way in the twelfth century:

Eminentiora ^{altiora} proluxarum ^{altarum} abietum sappinorum cacumina uel culmina uel uertices perindeque
dista ^{entia} acuto ^{claro} sonitu sono uel tono resultabant.

Let us also assume that it was copied in the thirteenth century into a new manuscript, *K*, as:

Eminentiora altarum sappinorum culmina perindeque distenta claro sonitu et tono resultabant.

A raw neighbour-joining calculation taking into account only the first hand of *B* would produce the following graph (fig. 5.5-11).

Due to contamination, *K* appears near the centre. If we take into account the variants and corrections added in *B*, with a concatenation of all of them, and then use a NeighborNet treatment, we obtain figure 5.5-12.

The graph is not easy to read, and would probably become unreadable if one added other witnesses with such a complex history. This kind of treatment of complex and non-tree-like traditions is only a stopgap solution and should be replaced, in this case, by more complex descriptions such as those presented by Andrews and Macé (2013). In short, material and evolutionary aspects of the manuscripts constitute the most important difference from the phylogenetic model and probably the most difficult challenge in digital stemmatology.

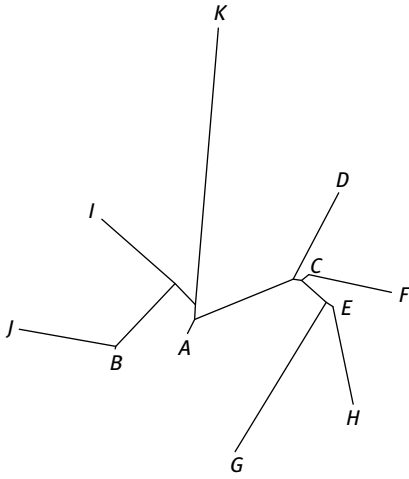


Fig. 5.5-11: Adding a new contaminated witness *K* to the above example.

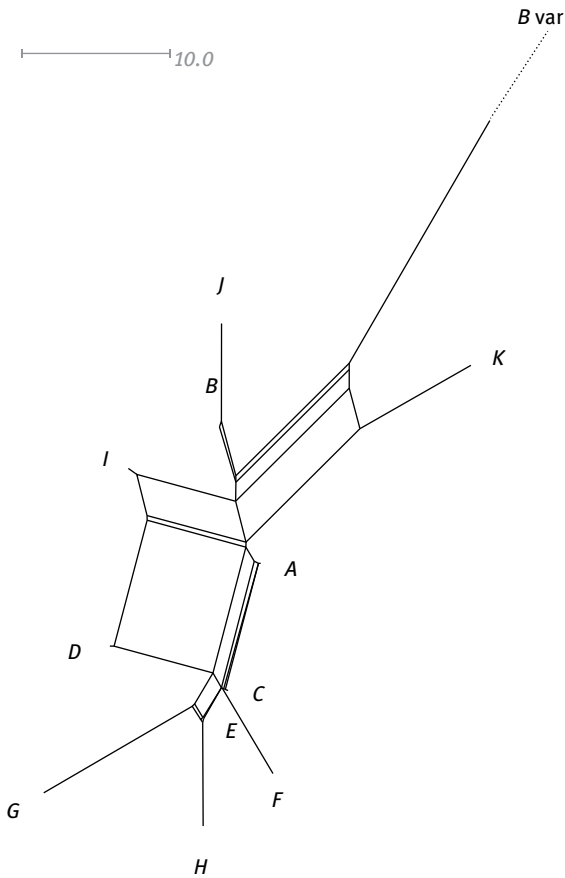


Fig. 5.5-12: NeighborNet graph for tradition with contaminated witness *K*. Plotted with SplitsTree4, with the branch length of *B var* reduced for presentation purposes.

5.5.10 Conclusion: For careful use guided by the philologist

To sum up this synthetic presentation of the different kinds of criticism that have sometimes been levelled against digital approaches in stemmatology, it is clear that algorithms in use for some twenty years have produced encouraging results, especially in allowing us to visualise quickly the most certain kinships in large textual traditions for which a traditional stemmatic approach would be much more tedious. However, intervention by the philologist still remains absolutely necessary to interpret the graphs produced by these methods: as has been said above, it would be absurd to consider them as stemmata. Indeed, the historical dimension of every textual tradition needs to be taken into account during the interpretation of the different graphs, even those obtained with methods specifically developed for stemmatology (such as RHM or Semstem). For every method, it thus remains necessary to apply *a posteriori* some transformations, especially concerning rooting, the polarisation of variants, or the detection of different forms of contamination.

Moreover, in order to become an efficient tool for philologists, digital stemmatology should allow philologists to work easily on raw data (e.g. semi-diplomatic transcriptions) without having to carry out long and tedious preparation (e.g. alignment tables). From this point of view, much has already been done on stemmaweb.net/stemmaweb, which provides the opportunity to try out several different tools. Finally, one could also expect the development of interaction between stemmatological software and digital editions: if all the variants of each witness of a tradition are encoded in a standard way (e.g. in TEI XML), it should be possible to adapt the available kinds of stemmatological software in order to make them work directly with this kind of base data. On the other hand, considering the improvements in character recognition (OCR), it could be exciting to envisage, in the medium term, a coupling of automatic transcription and stemmatic software in order to treat a larger amount of data.

As we have seen, digital stemmatology is still a very young discipline. Its main methods have been developed over the past twenty years. Despite its short history, and also despite the limitations presented above, this discipline should be regarded as a viable scientific auxiliary for textual criticism that embraces an interdisciplinary approach. In such cases, computer scientists, bioinformatic researchers, or digital humanities scholars and philologists should closely cooperate to ensure the validity and adequacy of their approaches. Obviously, such collaborative work should advance with respect for the expertise involved on all sides. And ultimately, as the domain experts, philologists need to decide on the historical plausibility and validity of the results. As careful interdisciplinary work, digital stemmatology in this way can contribute to the always necessary renewal of philology.