Compression-based Stemmatology:
A Study of the Legend of St. Henry of Finland

Teemu Roos[1], Tuomas Heikkilä[2],
Rudi Cilibrasi[3], and Petri Myllymäki[1]

[1] Helsinki Institute for Information Technology,
University of Helsinki and Helsinki University of Technology
[2] Department of History, University of Helsinki
[3] Centrum Wiskunde en Informatica (CWI), Amsterdam

28th November 2005

Abstract

Stemmatology studies relations among different variants of a text that has been gradually altered as a result of imperfectly copying the text over and over again. We propose a new computer-assisted method for stemmatic analysis based on compression of the variants. The method is related to phylogenetic reconstruction criteria such as maximum parsimony and maximum likelihood. We apply our method to the tradition of the legend of St. Henry of Finland, and report encouraging preliminary results. The obtained family tree of the variants, the stemma, corresponds to a large extent with results obtained with more traditional methods. Some of the identified groups of manuscripts are previously unrecognized ones. Moreover, due to the impossibility of manually exploring all plausible alternatives among the vast number of possible trees, this work is the first attempt at a complete stemma for the legend of St. Henry. The used methods are being released as open-source software.

I Introduction

ST. HENRY, according to the medieval tradition Bishop of Uppsala (Sweden) and the first Bishop of Finland, is the key figure of the Finnish Middle Ages. He seems to have been one of the leaders of a Swedish expedition to Finland probably around 1155. After this expedition Henry stayed in Finland with sad consequences: he was murdered already next year. He

1

Figure 1: An excerpt of a 15th century manuscript 'H' from the collections of the Helsinki University Library, showing the beginning of the legend of St. Henry on the right: *"Incipit legenda de sancto Henrico pontifice et martyre; lectio prima; Regnante illustrissimo rege sancto Erico, in Suecia, uenerabilis pontifex beatus Henricus, de Anglia oriundus, ..."* [12].

soon became the patron saint of Turku cathedral and of the bishopric covering the whole of Finland. He remained the only 'local' one of the most important saints until the reformation. Henry is still considered to be the Finnish national saint. The knowledge of writing was almost totally concentrated into the hands of the Church and the clergymen during the early and high Middle Ages. On the other hand, the official and proper veneration of a saint needed unavoidably a written text containing the highlights of the saint's life and an account of his miracles to be recited during the services in the church. The oldest text concerning St. Henry is his legend written in Latin. It contains both his life and a collection of his miracles and seems to have been ready by the end of the 13th century at the very latest. The text is the oldest literary work preserved in Finland and can thus be seen as the starting point of the Finnish literary culture. Whereas the influence of St. Henry on the Christianization of Finland has been one of the focusing points of the Finnish and Swedish medievalists for hundreds of years, only the most recent research has really concentrated on his legend as a whole. According to the latest results, the Latin legend of St. Henry is known in 52 different medieval versions preserved in manuscripts and incunabula written in the early 14th–early 16th centuries (Fig. 1).[1]

The reasons for such a substantial amount of versions differing from each

---

[1]For identification of the sources as well as a modern edition of the legend see [12].

other are several. On one hand, the texts were copied by hand until the late 15th and early 16th centuries, which resulted in a multitude of unintended scribal errors by the copyists. In addition, the significance of the cult of St. Henry varied considerably from one part of the Latin Christendom to the other. In the medieval bishopric of Turku covering the whole of medieval Finland St. Henry was venerated as the most important local saint, whose adoration required the reciting of the whole legend during the celebrations of the saint's day. In Sweden, for instance, St. Henry was not so important a saint, which led to different kinds of abridgements fitted into the needs of local bishoprics and parishes. As a consequence, the preserved versions of the legend are all unique.

With the aid of traditional historically oriented auxiliary sciences like codicology and paleography it is possible to find out — at least roughly — where and when every version was written. Thus, the versions form a pattern representing the medieval and later dissemination of the text. Even if the existent manuscripts containing the different versions represent but a tiny part of the much larger number of manuscripts and versions written during the Middle Ages, they still provide us with an insight into a variety of aspects of medieval culture. The versions help to reconstruct the actual writing process and the cultural ties that carried the text from one place to another. When one combines the stemma — i.e. the family tree — of a text with a geographical map and adds the time dimension, one gets important information that no single historical source can ever provide a historian with. The potential of this kind of an approach is emphasized when researching hagiographical texts — i.e. saints' lives, for instance — since they were the most eagerly read and most vastly disseminated literary genre of the Middle Ages.

Taking into consideration the possibilities of stemmatology, it is not surprising that the historians and philologists have tried to establish a reliable way to reconstruct the stemma of the text and its versions for centuries. The main difficulty has been the great multitude of textual variants that have to be taken into consideration at the same time. An example from the legend material of St. Henry shall elucidate the problems: there are over 50 manuscripts and incunabula to be taken into consideration; in the relatively short text there are nearly one thousand places where the versions differ from each other. Since the multitude of variants rises easily to tens of thousands, it has been impossible for researchers using traditional methods of paper and pen to form the stemma and thus get reliable answers to the questions related to the writing and disseminating of the text. There have been some previous attempts to solve the problems of stemmatology with

3

the aid of computer science. In addition, the powerful computer programs developed for the needs of the computer-aided cladistics in the field of evolutionary biology have been used. In many cases this has proven to be a fruitful approach, extending the possibilities of stemmatics to the analysis of more complex textual traditions that are outside the reach of manual analysis. Moreover, formalizing the often informal and subjective methods used in manual analysis makes the methods and results obtained with them more transparent and brings them under objective scrutiny. Still, many issues in computer-assisted stemmatic analysis remain unsolved, underlining the importance of advances towards general and reliable methods for shaping the stemma of a text.

The paper is organized as follows: In Sec II we present a criterion for stemmatic analysis that is based on compression of the manuscripts. We then outline an algorithm, in Sec. III, that builds stemmata by comparing a large number of tree-shaped stemmata and choosing the one that minimizes the criterion. The method is demonstrated on a simple example in Sec. IV, where we also present our main experiment using some 50 variants of the legend of St. Henry, and discuss some of the restrictions of the method and potential ways to overcome them. Conclusions are presented in Sec. V. We also compare our method to a related method in the CompLearn package in Appendix A.

II   A Minimum-Information Criterion

One of the most applied methods in biological phylogeny is maximum parsimony. A maximally parsimonious tree minimizes the total number of differences between connected nodes — i.e., species, individuals, or manuscripts that are directly related — possibly weighted by their importance. In stemmatology, analysis is based on variable readings that result from unintentional errors in copying or intentional omissions, insertions, or other modifications. In his seminal work on computer-assisted stemmatology, O'Hara used a parsimony method of the PAUP software [24] in Robinson's Textual Criticism challenge [20]. For further applications of maximum parsimony and related method, see [13, 16, 23, 26] and references therein.

Our compression-based *minimum information* criterion shares many properties of the maximum parsimony method. Both can also be seen as instances of the *minimum description length* (MDL) principle of Rissanen [19] — although this is slightly anachronistic: the maximum parsimony method predates the more general MDL principle — which in turn is a formal version of Ockham's razor. The underlying idea in the minimum information

4

criterion is to minimize the amount of information, or *code-length*, required to reproduce all the manuscripts by the process of copying and modifying the text under study. In order to describe a new version of an existing manuscript, one needs an amount of information that depends on both the amount and the type of modifications made. For instance, a deletion of a word or a change of word order requires less information to describe compared to introducing a completely new expression. In order to be concrete, we need a precise, numerical, and computable measure for the amount of information. The commonly accepted definition of the amount information in individual objects is Kolmogorov complexity [14, 17], defined as the length of the shortest computer program to describe the given object. However, Kolmogorov complexity is defined only up to a constant that depends on the language used to encode programs, and what is more, fundamentally uncomputable. In the spirit of a number of earlier authors [1, 3, 4, 6, 11, 18, 25] we approximate Kolmogorov complexity by using a compression program. Currently, we use `gzip` based on the LZ77 [27] algorithm, and plan to experiment with other compressors in subsequent work. In particular, given two strings, x and y, the amount of information in y conditional on x, denoted by $C(y \mid x)$ is given by the length of the compressed version of the concatenated string x, y minus the length of the compressed version of x alone[2]. A simple example illustrating these concepts is given below in Sec. IV.

In addition to the MDL interpretation, our method can be seen as (an approximation of) maximum likelihood, another commonly used criterion in phylogeny. The maximum likelihood criterion requires that we have a probabilistic model for evolution, assigning specific probabilities for each kind of change. The joint likelihood of the whole tree is then evaluated as a product of likelihoods of the individual changes. The tree achieving the highest joint likelihood given the observed data is then preferred. In the case of manuscripts such a model is clearly more difficult to construct that in biology, where the probabilities of mutation can be estimated from experimental data. Nevertheless, a model for manuscript evolution is presented in [22]. Code-length is isomorphic to (behaves in the same way as) likelihood: sums of code-lengths have a direct correspondence with products of likelihoods. If the probability induced by the information cost, $2^{-C(y|x)}$, is approximately proportional to the likelihood of creating a copy y based on the original x, then minimizing the total information cost approximates maximizing the likelihood.

Let $G = (V, E)$ be an undirected graph where V is a set of nodes corres-

---

[2]We insert a newline in the end of each string and between x and y.

ponding to the text variants, $E \subset V \times V$ is a set of edges. We require that the graph is a connected bifurcating tree, i.e., that (i) each node has either one or three neighbors, and (ii) the tree is acyclic. Such a graph G can be made directed by picking any one of the nodes as a root and directing each edge away from the root. Given a directed graph $\vec{G}$, the total information cost of the tree is given by

$$C(\vec{G}) = \sum_{v \in V} C(v \mid \mathrm{Pa}(v))$$

$$= \sum_{v \in V} C(\mathrm{Pa}(v), v) - C(\mathrm{Pa}(v)), \tag{1}$$

where $\mathrm{Pa}(v)$ denotes the parent node of v unless v is the root in which case $\mathrm{Pa}(v)$ is the empty string. Assuming that order has no significant effect on the complexity of a concatenated string, i.e., we have $C(x,y) \approx C(y,x)$, as seems to be the case in our data, it can easily verified that for acyclic bifurcating trees, the above can rewritten as

$$C(G) \approx \sum_{(v,w) \in E} C(v,w) - 2 \sum_{v \in V_I} C(v), \tag{2}$$

where the first summation has a term for each edge in the graph, and the second summation goes over the set of interior nodes V_I. The formula is a function of the undirected structure G only: the choice of the root is irrelevant. The factor two in the latter term comes from using *bi*furcating trees.

For practical reasons we make three modifications to this criterion. First, as we explain in the next section, due to algorithmic reasons we need to splice the texts in smaller segments, not longer than roughly 10–20 words (we used 11). Secondly, we found that the cost assigned by `gzip` to reproducing an identical copy of a string is too high in the sense that it is sometimes 'cheaper' to omit a large part of the text for a number of generations and to re-invent it later in an identical form. Therefore we define the cost of making an identical copy to be zero. Thirdly, it is known that the variation between an ampersand ('&') and the word *et*, and the letters v and u was mostly dependent on the style of the copyist and changed with time and region, and thus, bears little information relevant to stemmatic analysis. This domain knowledge was taken into account by replacing, in both of the above cases, all occurrences of the former by the latter[3]. Thus, we use the following

---

[3]Howe *et al.* [13] use as an example the words *kirk* and *church* in 15th century English whose variation mainly reflects local dialect.

modified cost function

$$C'(\vec{G}) = \sum_{v \in V} \sum_{i=1}^{n} C'(v_i \mid \mathrm{Pa}_i(v)), \qquad (3)$$

where n is the number of segments into which each text is spliced, v_i and $\mathrm{Pa}_i(v)$ are the ith segment of variant v and its parent, respectively, all strings are modified according to the above rules (ampersand to et, and v to u), and $C'(x \mid y)$ equals the `gzip` cost if x and y differ, and zero otherwise. This modified cost also allows a form similar to (2) and hence, is practically independent of the choice of the root.

III An Algorithm for Constructing Stemmata

Since it is known that many of the text variants have been lost during the centuries between the time of the writing of the first versions and present time, it is not realistic to build a tree of only the about 50 variants that we have as our data. This problem is even more prominent in biology where we can only make observations about organisms that still exist (excluding fossil evidence). The common way of handling this problem is to include in the tree a number of 'hidden' nodes, i.e., nodes representing individuals whose characteristics are unobserved. We construct bifurcating trees that have N observed nodes as leafs, and $N-2$ hidden nodes as the interior nodes.

Evaluating the criterion (3) now involves the problem of dealing with the hidden nodes. Without knowing the values of $\mathrm{Pa}_i(v)$, it is not possible to compute $C'(v \mid \mathrm{Pa}_i(v))$. We solve this problem by searching simultaneously for the best tree structure $\vec{G}$ and for the optimal contents of the hidden nodes with respect to criterion (3). As mentioned above, we patch up the contents of the interior nodes from segments of length 10–20 words appearing in some of the available variants. In principle we would like to do this on a per-word-basis, which would not be a notable restriction since it is indeed reasonable to expect that a reconstruction only consists of words appearing in the available variants — any other kind of behavior would require rather striking innovation. However, since we evaluate the `gzip` cost in terms of the segments, it is likely give better values when the segments are longer than one word. Secondly, one of the most common modifications is change in word order. Using 10-20 word segments we assign less cost to change in word order than to genuine change of words, unless the change happens to cross a segment border.

Perhaps surprisingly, given a tree structure, finding the optimal contents is feasible. The method for efficiently optimizing the contents of the hidden

7

nodes is an instance of dynamic programming and called 'the Sankoff algorithm' [8] or 'the Felsenstein's algorithm' [21]. As Siepel and Haussler [21] note, it is in fact an instance of a 'message-passing' or 'elimination' algorithm in graphical models (see also [10]). The basic idea is to maintain for each node a table of minimal costs for the whole subtree starting at the node, given that the contents of the node take any given value. For instance, let us fix a segment, and denote by $x^1, \ldots, x^m$ the different versions of the segment that appear in some of the observed variants. The minimal cost for the subtree starting at node i, given that the segment in question of node i contains the string x^j is given by (see [8])

$$\text{cost}_i(j) = \min_k \left[C'(x^k \mid x^j) + \text{cost}_a(k) \right] + \min_l \left[C'(x^l \mid x^j) + \text{cost}_b(l) \right],$$

where a and b are the two children of node i. For leaf nodes the cost is defined as being infinite if j does not match the known content of the node, and zero if j matches or if the content of the node is unknown. Evaluating $\text{cost}_i(j)$ can be done for each segment independently, starting from the leaf nodes and working towards the root. Finally, the (unconditional) complexity of the root is added so that the minimal cost of the segment is obtained by choosing at the root the string x^j that minimizes the sum $\text{cost}_{\text{root}}(j) + C'(x^j)$. The total cost of the tree is then obtained by summing over the minimal costs for each segment. After this, actually filling the contents can be done by propagating back down from the root towards the leafs. It is important to remember that while the algorithm for optimizing the contents of the hidden nodes requires that a root is selected, the resulting cost and the optimal contents of the hidden nodes only depend on the undirected structure (see Eq. (2)).

There still remains the problem of finding the tree structure, which together with corresponding optimal contents of the hidden nodes minimizes criterion (3). The obvious solution, trying all possible tree structures and choosing the best one, fails because for N leafs nodes, the number of possible bifurcating trees is as large as (see [8])

$$1 \times 3 \times 5 \times \ldots \times (2N - 5).$$

For $N = 52$ this number is about 2.73×10^{78}, which is close to the estimated number of atoms in the universe. Instead, we have to resort to heuristic search, trying to find as good a tree as possible in the time available.

We use a simulated annealing algorithm which starts with an arbitrary tree and iteratively tries to improve it by small random modification, such

as exchanging the places of two subtrees[4]. Every modification that reduces the value of the criterion is accepted. In order to escape local optima in the search space, modifications that increase the value are accepted with probability

$$\exp\left(\frac{C'_{\text{old}} - C'_{\text{new}}}{T}\right),$$

where C'_{old} is the cost of the current tree, C'_{new} is the cost of the modified tree, and T is a 'temperature' parameter that is slowly decreased to zero. In our main experiment, reported in the next section, we ran 1,200,000 iterations of annealing, which we found to be sufficient in our setting.

IV   Results and Discussion

We first illustrate the behavior of the method by an artificial example in Fig. 2. Assume that we have observed five pieces of text, shown at the tips of the tree's branches. Because the text is so short, the length of the segment was fixed to one word. One of the trees — not the only one — minimizing the information cost with total cost of 44 units (bytes) is drawn in the figure. Even though, as explained above, the obtained tree is undirected, let us assume for simplicity that the original version is the topmost one (*"sanctus henricus ex Anglia"*). The sum of the (unconditional) complexities of the four words in this string is equal to $8 + 9 + 3 + 7 = 27$, which happens to coincide with the length of the string, including spaces and a finishing newline. The changes, labeled by number 1–5 in the figure, yield $5 + 3 + 3 + 3 + 3 = 17$ units of cost. Thus the total cost of the tree equals $27 + 17 = 44$ units.

As our main experiment, we analyzed a set of 49 variants of the legend of St. Henry. We had prepared four out of the nine sections (sections 1,4,5, and 6) in a suitable format. Three variants were excluded since they had only ten words or less in the prepared sections. The remaining variants contained 33–379 words each. Table V on page 17 lists the estimated time or writing and place of origin, as well as the number of words in the used sections for each manuscript. The best (wrt. the information cost) tree found is shown in Fig. 3. By comparing the tree with earlier results [12], it can be seen that many groups of variants have been successfully placed next to each other. For instance, groups of Finnish variants appearing in the tree that are believed to be related are Ho–I–K–T and R–S. Among the printed versions the pairs BA–BS and BLu–BL are correctly identified and also grouped close the each

---

[4]The algorithm also takes advantage of the fact that changes like exchanging subtrees only require partial updating of the dynamic programming table used to evaluate the information cost.

sanctus henricus ex Anglia

1.

beatus henricus ex Anglia

beatus henricus ex Anglia   beatus henricus ex Anglia

2.   *3. & 4.*

beatus henricus *in* Anglia   beatus *Henricus* ex *anglia*

5.

beatus Henricus *in* anglia

beatus Henricus ex anglia

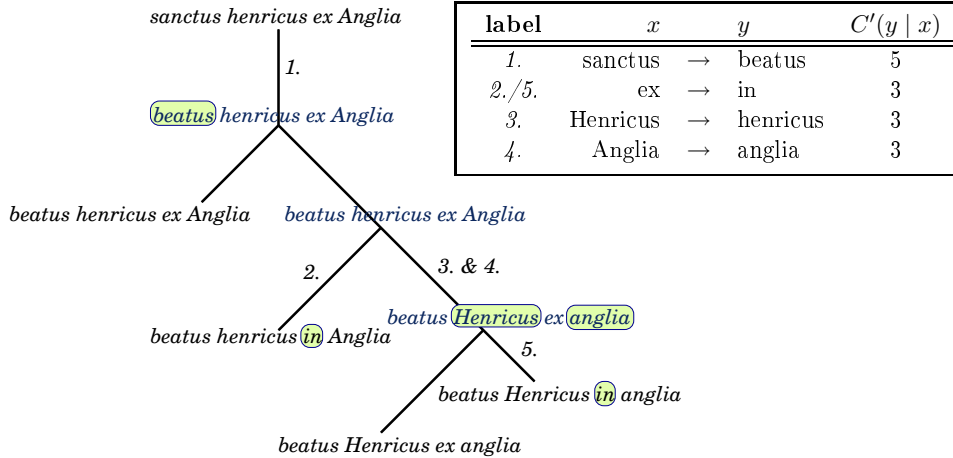| label	x		y	$C'(y \mid x)$
1.	sanctus	$\rightarrow$	beatus	5
2./5.	ex	$\rightarrow$	in	3
3.	Henricus	$\rightarrow$	henricus	3
4.	Anglia	$\rightarrow$	anglia	3

Figure 2: An example tree obtained with the compression-based method. Changes are circled and labeled with numbers 1–5. Costs of changes are listed in the box. Best reconstructions at interior nodes are written at the branching points.

other[5]. Other pairs of variants appearing in the tree that are believed to be directly related are Li–Q (that are also correctly associated with BA–BS and BL–BLu), JG–B, Dr–M, NR2–JB, LT–E, AJ–D, and Bc–MN–Y. In addition, the subtree including the nine nodes between (and including) BU and Dr is rather well supported by traditional methods. All in all, the tree corresponds very well with relationships discovered with more traditional methods. This is quite remarkable taking into account that in the current experiments we have only used four out of the nine sections of the legend.

In order to quantify confidence in the obtained trees we used on top of our method, block-wise bootstrap [15] and a consensus tree program in the phylogeny inference package Phylip [9]. One hundred bootstrap samples were generated by sampling (with replacement) n segments out of the n segments that make each manuscript. The compression-based method described in this work was run on each bootstrap sample — this took about a week of computation — and the resulting 100 trees were analyzed with the `consense` program in Phylip using default settings (modified majority rule). The resulting consensus tree is shown in Fig. 4.

It should be noted that the central node with nine neighbors does not corresponds to a single manuscript with nine descendants, but rather, that

---

[5]The printed versions are especially suspect to contamination since it is likely that more than one manuscript was used when composing a printed version.

Figure 3: Best tree found. Most probable place of origin according to [12], see Table V, indicated by color — Finland (blue): K,Ho,I,T,A,R,S,H,N,Fg; Vadstena (red): AJ,D,E,LT,MN,Y,JB,NR2,Li,F,G; Central Europe (yellow): JG,B; other (green). Some groups supported by earlier work are circled in red.

the relationships between the nine subtrees is unidentified. Because the interpretation of the consensus tree is less direct than the interpretation of the tree in Fig. 3 as the family tree of the variants, it is perhaps best to use the consensus tree to quantify the confidence in different parts of the tree in Fig. 3. For instance, it can be seen that the pairs BL–BLu, AJ–D, Li–Q, NR2–JB, O–P, L–G, JG–B, and R–S are well supported. More interestingly, The group Ho–I–K–T–A is organized in a different order in Fig. 3 and the consensus tree. This group also illustrates one of the problems

11

in the consensus tree method. Namely the confidence in contiguous groups that are in the middle of the tree tends to be artificially low since the group does not make up a subtree, in this case only 3/100 (Fig. 4).

The following potential problems and sources of bias in the resulting stemmata are roughly in decreasing order of severity:

1. The `gzip` algorithm does not even attempt to fully reflect the process of imperfectly copying manuscripts. It remains to be studied how sensible the `gzip` information cost, or costs based on other compression algorithms, are in stemmatic analysis.

2. Trees are not flexible enough to represent all realistic scenarios. More than one original manuscript may have been used when creating a new one — a phenomenon termed *contamination* (or horizontal transfer in genomics). Point 5 below may provide a solution but for non-tree structures the dynamic programming approach doesn't work and serious computational problems may arise.

3. Patching up interior node contents from 10–20 word segments is a restriction. This restriction could be removed for cost functions that are defined as a sum of individual words' contributions. Such cost functions may face problems in dealing with change of word order.

4. The number of copies made from a single manuscript can be other than zero and two. The immediate solution would be to use multifurcating trees in combination with our method, but this faces the problem that the number of internal nodes strongly affects the minimum-information criterion. The modification hinted to at point 5 may provide a solution to this problem.

5. Rather than looking for the tree structure that together with the optimal contents of the interior nodes minimizes the cost, it would be more principled from a probabilistic point of view to 'marginalize' the interior nodes (see [10]). In this case we should also account for possible forms (words or segments) not occurring in any of the observed variants.

6. The search space is huge and the algorithm only finds a local optimum whose quality cannot be guaranteed. Bootstrapping helps to identify which parts of the tree are uncertain due to problems in search (as well as due to lack of evidence).

7. Bootstrapping is known to underestimate the confidence in the resulting consensus tree. This is clearly less serious than *over*estimation.

In future work we plan to investigate ways to overcome some of these limitations, to carry out more experiments with more data in order to validate the method and to compare the results with those obtained with, for instance, the existing methods in CompLearn [5], Phylip [9], and PAUP [24]. We are also planning to release the software as a part of the CompLearn

package. Among the possibilities we have not yet explored is the reconstruction of a likely original text. In fact, in addition to the stemma, the method finds an optimal — i.e., optimal with respect to the criterion — history of the manuscript including a text version at each branching point of the stemma. Assuming a point of origin, or a root, in the otherwise undirected stemma tree, thus directly suggests a reconstruction of the most original version.

V Conclusions

We proposed a new compression-based criterion, and an associated algorithm for computer-assisted stemmatic analysis. The method was applied to the tradition of the legend of St. Henry of Finland, of which some fifty manuscripts are known. Even for such a moderate number, manual stemma reconstruction is prohibitive due to the vast number of potential explanations, and the obtained stemma is the first attempt at a complete stemma of the legend of St. Henry. The relationships discovered by the method are largely supported by more traditional analysis in earlier work, even though we have thus far only used a part of the legend in our experiments. Moreover, our results have pointed out groups of manuscripts not noticed in earlier manual analysis. Consequently, they have contributed to research on the legend of St. Henry carried out by historians and helped in forming a new basis for future studies. Trying to reconstruct the earliest version of the text and the direction of the relationships between the nodes in the stemma is an exciting line of research where a combination of stemmatological, palaeographical, codicological and contentual analysis has great potential.

Appendix A: Comparison with the CompLearn package

The CompLearn package [5] performs similar analysis as our method in a more general context where the strings need not consist of word-by-word aligned text. It is based on the Normalized Compression Distance (NCD) defined as

$$\text{NCD}(x, y) = \frac{\max\{C(x \mid y), C(y \mid x)\}}{\max\{C(x), C(y)\}}, \tag{4}$$

that was developed and analyzed in [2, 3, 4, 6, 17]. Both our minimum information criterion and NCD are based on (approximations of) Kolmogorov complexity. The core method in CompLearn uses a quartet tree heuristic in order to build a bifurcating tree with the observed strings as leafs [7]. In contrast to our method, where the cost function involves the contents of both

13

the observed strings in the leaves and the unobserved interior nodes, Comp-Learn only uses the pairwise NCD distances between the observed strings (in [8] the latter kind of methods are called distance matrix methods).

The relation between NCD and the criterion presented in this work may be made more clear by considering the sum-distance $C(y \mid x) + C(x \mid y)$. Bennett *et al.* [2] show that the sum-distance is sandwiched between the numerator of (4) and two times the same quantity, ignoring logarithmic terms:

$$\max\{C(x \mid y), C(y \mid x)\} \leq C(y \mid x) + C(x \mid y) \leq 2\max\{C(x \mid y), C(y \mid x)\}. \tag{5}$$

Assuming that $C(x, y) \approx C(y, x)$ for all x, y, the sum-distance yields the cost

$$\sum_{(v,w)\in E} C(w \mid v) + C(v \mid w) = 2 \sum_{(v,w)\in E} C(v, w) - 3 \sum_{v\in V_I} C(v) - \sum_{w\in V_L} C(w),$$

where the summations are over the set of edges E, the set of interior nodes V_I, and the set of leaf nodes V_L, respectively. Since the set of leaf nodes is constant in the phylogenetic reconstruction problem, the last term can be ignored. Comparing the first two terms with (2) shows that the only difference is in the ratio of the factors of the first two terms (2 : 3 above; 1 : 2 in (2)). Thus, the difference between the the sum-distance and the information cost depends only on the variation of $C(v)$: if all strings are of roughly the same complexity, the difference is small. On the other hand, the difference between the sum-distance and NCD results, up to a factor of two (inequality (5)), from the normalization by $\max\{C(x), C(y)\}$ in (4). Thus, if all strings are equally complex, the sum-distance and NCD do not differ 'too much', which in turn implies, *summa summarum*, that the information cost and NCD agree, at least roughly. However, in our case, many of the variants are partially destroyed, and consequently the complexity of the existing texts varies. The difference between the quartet tree heuristic and our Sankoff-style algorithm (Sec. III) is more difficult to analyze, but clearly, both are designed for the same purpose.

Figure 5 shows the tree obtained by CompLearn using a blocksort approximation to Kolmogorov complexity (see the documentation of CompLearn for more information). The tree agrees at least roughly in many places with the tree in Fig. 3, for instance, the expected pairs Ho–T, JB–NR2, D–AJ, JG–B, MN–Y, BA–BS, and LT–E are next to or almost next to each other in both trees. We plan to investigate whether the remaining differences between the two trees are due to the cost functions, the search methods, or other features

14

of the methods. At any rate, such agreements corroborate the validity of both methods and provide yet stronger support for the results.

Acknowledgments

References

[1] D. Benedetto, E. Caglioti, and V. Loreto. Language trees and zipping. *Physical Review Letters*, 88(4):048702–1–048702–4, 2002.

[2] C.H. Bennett, P. Gacs, M. Li, P.M.B. Vitányi, and W. Zurek. Information distance. *IEEE Transactions on Information Theory*, 44(4):1407–1423, 1998.

[3] C.H. Bennett, M. Li, and B. Ma. Chain letters and evolutionary histories. *Scientific American*, pages 76–81, November 2003.

[4] X. Chen, S. Kwong, and M. Li. A compression algorithm for DNA sequences and its applications in genome comparison. In K. Asai, S. Miyano, and T. Takagi, editors, *Genome Informatics*, Tokyo, 1999. Universal Academy Press.

[5] R. Cilibrasi, A.-L. Cruz, and S. de Rooij. Complearn version 0.8.20, 2005. Distributed at www.complearn.org.

[6] R. Cilibrasi and P.M.B. Vitányi. Clustering by compression. *IEEE Transactions on Information Theory*, 51(4):1523–1545, 2005.

[7] R. Cilibrasi and P.M.B. Vitányi. A new quartet tree heuristic for hierarchical clustering. In *EU-PASCAL Statistics and Optimization of Clustering Workshop*, London, 2005.

[8] J. Felsenstein. *Inferring phylogenies*. Sinauer Associates, Sunderland, Massachusetts, 2004.

[9] J. Felsenstein. PHYLIP (Phylogeny inference package) version 3.6, 2004. Distributed by the author, Department of Genome Sciences, University of Washington, Seattle.

[10] N. Friedman, M. Ninio, I. Pe'er, and T. Pupko. A structural EM algorithm for phylogenetic inference. *Journal of Computational Biology*, 9:331–353, 2002.

[11] S. Grumbach and F. Tahi. A new challenge for compression algorithms: genetic sequences. *Journal of Information Processing and Management*, 30(6):875–866, 1994.

[12] T. Heikkilä. *Pyhän Henrikin legenda* (in Finnish). Suomalaisen Kirjallisuuden Seuran Toimituksia 1039, Helsinki, 2005.

[13] C.J. Howe, A.C. Barbrook, M. Spencer, P. Robinson, B. Bordalejo, and L.R. Mooney. Manuscript evolution. *Trends in Genetics*, 17(3):147–152, 2001.

[14] A.N. Kolmogorov. Three approaches to the quantitative definition of information. *Problems in Information Transmission*, 1(1):1–7, 1965.

[15] H.R. Künsch. The jackknife and the bootstrap for general stationary observations. *Annals of Statistics*, 17(3):1217–1241.

[16] A.-C. Lantin, P. V. Baret, and C. Macé. Phylogenetic analysis of Gregory of Nazianzus' Homily 27. In G. Purnelle, C. Fairon, and A. Dister, editors, *7èmes Journées Internationales d'Analyse statistique des Données Textuelles*, pages 700–707, Louvain-la-Neuve, 2004.

[17] M. Li and P.M.B. Vitányi. *An Introduction to Kolmogorov Complexity and Its Applications, 2nd. Ed.* Springer-Verlag, New York, 1997.

[18] D. Loewenstern, H. Hirsh, P. Yianilos, and M. Noordewier. DNA sequence classification using compression-based induction. Technical Report 95–04, DIMACS, 1995.

[19] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

[20] P. Robinson and R.J. O'Hara. Report on the textual criticism challenge 1991. *Bryn Mawr Classical Review*, 3(4):331–337, 1992.

[21] A. Siepel and D. Haussler. Phylogenetic estimation of context-dependent substitution rates by maximum likelihood. *Molecular Biology and Evolution*, 21(3):468–488, 2004.

[22] M. Spencer and C.J. Howe. How accurate were scribes? A mathematical model. *Literary and Linguistic Computing*, 17(3):311–322, 2002.

[23] M. Spencer, K. Wachtel, and C.J. Howe. The Greek Vorlage of the Syra Harclensis: A comparative study on method in exploring textual genealogy. *TC: A Journal of Biblical Textual Criticism*, 7, 2002.

[24] D.L. Swofford. PAUP*: Phylogenetic analysis using parsimony (*and other methods). version 4., 2003.

[25] J.-S. Varre, J.-P. Delahaye, and É. Rivals. The transformation distance: a dissimilarity measure based on movements of segments. In *Proceedings of German Conference on Bioinformatics*, Koel, Germany, 1998.

[26] E. Wattel and M.P. van Mulken. Weighted formal support of a pedigree. In P. van Reenen and M.P. van Mulken, editors, *Studies in Stemmatology*, pages 135–169. Benjamins Publishing, Amsterdam, 1996.

[27] J. Ziv and A. Lempel. A universal algorithm for sequential data compression. *IEEE Transactions on Information Theory*, 23(3):337–343, 1977.

Table 1. Estimated time of writing and place of origin (alternative place in parentheses) from [12], and total number of words in Secs. 1,4,5, and 6.

| Code	Time	Place	# of Words
A	1st half of 14th c.	Finland (/Sweden)	364
Ab	14th c.	Finland	7
AJ	1416–1442	Vadstena	185
B	ca. 1460	Cologne	336
BA	1513	Västerås	185
Bc	15th c.	Sweden	250
BL	1493	Linköping	246
BLu	1517	Lund	185
BS	1498	Skara	185
BSt	1495	Strängnäs	189
BU	1496	Uppsala	329
C	14th to 15th c.	Sweden	375
Cd	15th c.	Sweden (/Finland)	102
CP	1462–1500	Vadstena	59
D	1446–1460	Vadstena	181
De	15th c.	Växjö (/Sweden)	95
Dr	end of 14th c.	Linköping (/Växjö)	371
E	1442–1464	Vadstena	237
Ef	end of 14th c. / .... beginning of 15th c.	Sweden (/Finland)	82
F	1st half of 15th c.	Vadstena (/Linköping)	339
Fg	14th c.	Finland (Sweden)	44
G	1476–1514	Vadstena	251
Gh	14th c.	Sweden (/Finland)	97
H	end of 14th c. / .... beginning of 15th c.	Finland	74
Ho	after 1485	Hollola	371
I	end of 15th c. / .... beginning of 16th c.	Ikaalinen	267
JB	1428–1447	Vadstena	166
JG	ca. 1480	Brussels	341
K	end of 15th c. / .... beginning of 16th c.	Kangasala	372
L	15th c.	Sweden	132
Li	2nd half of 15th c.	Vadstena	193
LT	1448–1458	Vadstena	266
Lu	1st half of 14th c.	Sweden	149
M	1st half of 15th c.	Bishopric of Linköping	228
MN	1495	Vadstena	372
N	15th c.	Finland	373
NR	1476–1514	Vadstena	0
NR2	after 1489	Vadstena	158
O	middle 14th c.	Ösmo (/Uppsala)	182
P	ca. 1380	Strängnäs (/Vadstena)	379
Q	2nd half of 15th c., ... before 1493	Bishopric of Linköping .... (/Vadstena)	176
R	15th c.	Finland	267
S	1st half of 15th c.	Finland	370
St	beginning of 15th c.	Bishopric of Strängnäs (/Sweden) ..	211
T	ca. 1485	Finland	373
U	15th c.	Uppsala	154
V	1485	Bishopric of Uppsala	301
Vae	14th c.	Sweden (/Finland)	247
Vg	end of 14th c. / .... beginning of 15th c.	Sweden (/Finland)	33
X	middle or late 15th c.	Bishopric of Uppsala	188
Y	ca. 1500	Vadstena (/Linköping)	372
Z	15th c.	Sweden (/Finland)	10
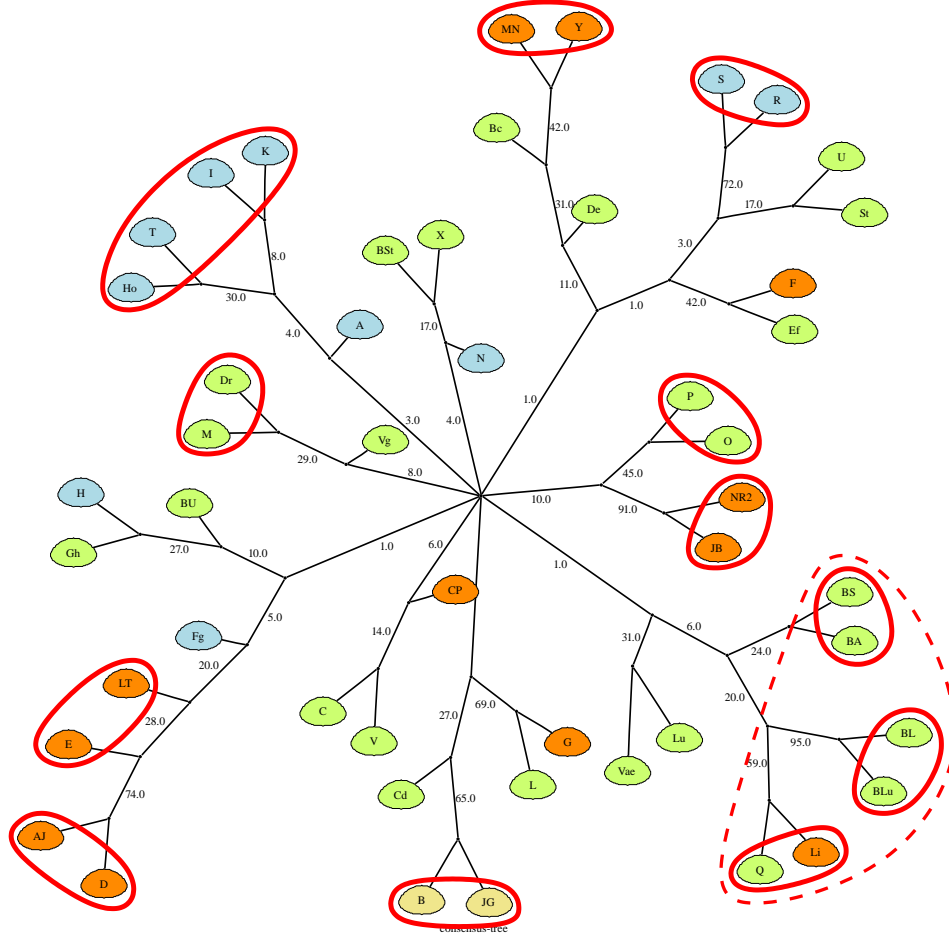
Figure 4: Consensus tree. The numbers on the edges indicate the number of bootstrap trees out of 100 where the edge separates the two sets of variants. Large numbers suggest high confidence in the identified subgroup. Some groups supported by earlier work are circled in red.
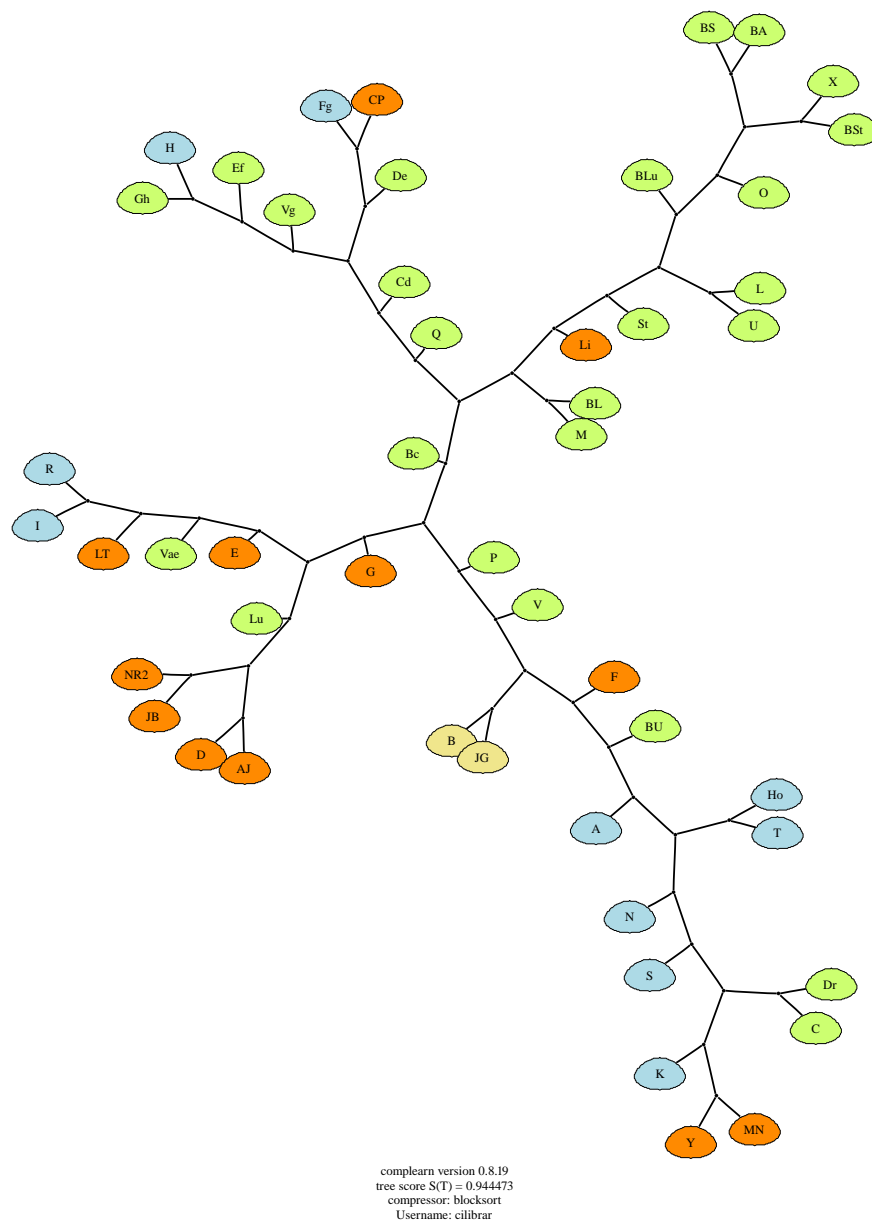
complearn version 0.8.19
tree score S(T) = 0.944473
compressor: blocksort
Username: cilibrar

Figure 5: CompLearn tree showing many similarities with the tree in Fig. 3.