A Compression-Based Method for Stemmatic Analysis[1]

Teemu Roos [2] and **Tuomas Heikkilä** [3] and **Petri Myllymäki** [1]

May 10, 2006

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. Applications are mainly in humanities, especially textual criticism, but the methods can be used to study the evolution of any symbolic objects, including chain letters and computer viruses.We propose an algorithm for stemmatic analysis based on a minimum-information criterion and stochastic tree optimization. Our approach is related to phylogenetic reconstruction criteria such as maximum parsimony and maximum likelihood, and builds upon algorithmic techniques developed for bioinformatics. Unlike many earlier methods, the proposed method does not require significant preprocessing of the data but rather, operates directly on aligned text files. We demonstrate our method on a real-world experiment involving all 52 known variants of the legend of St. Henry of Finland, and provide the first computer-generated family tree of the legend. The obtained tree of the variants is supported to a large extent by results obtained with more traditional methods, and identifies a number of previously unrecognized relations.

1 INTRODUCTION

We begin with a brief historical motivation to the problem under study. During the early and high Middle Ages, the knowledge of writing was almost totally concentrated into the hands of the Church and the clergymen. Hagiographical texts, i.e. texts concerning saints' lives were the most eagerly read and most vastly disseminated literary genre. In particular, the official and proper veneration of a saint needed unavoidably a written text, a legend, containing the highlights of the saint's life. In the case of most legends, the text itself has survived to our date in several different versions. Underlying these versions there is what we could call a 'family tree', a graph representing the process of copying the text where each new version becomes a direct descendant of the exemplar(s) from which it is copied. The aim of stemmatic analysis is to reconstruct this family tree, known as the 'stemma', based on the surviving copies of the text. By studying the materials and the writing of the available versions, it is possible to find out — at least roughly — where and when each version was written. When one combines the stemma 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: a reconstruction of the process of dissemination, and the cultural ties that carried the text from one place to another.

The reasons for a substantial amount of versions differing from each 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 saints' cults varied considerably from one part of the Latin Christendom to the other. The adoration of the most important local saints required the reciting of the whole legend during the celebrations of the saint's day. On the other hand, in cases of lesser importance, different kinds of abridgements were fitted into the needs of local bishoprics and parishes. As a consequence, the preserved versions of most legends are all unique.

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. A related application is the analysis of chain letters [2]. 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[4] elucidates the problems. According to latest knowledge, the Latin legend of St. Henry is known in 52 different medieval versions[5] preserved in manuscripts and incunabula (early printed works) written in the early 14th–early

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, ..."* [10].

[1] This is an extended version of a two-page summary with the same title to appear in *Proceedings of the 17th European Conference on Artificial Intelligence* (ECAI'06), Riva del Garda, Italy, August–September, 2006.

[2] Complex Systems Computation Group, Helsinki Institute for Information Technology, Finland, email: firstname.lastname@cs.helsinki.fi

[3] Dept. of History, University of Helsinki, Finland, email: tuomas.m.heikkila @helsinki.fi

[4] St. Henry is a key figure of the Finnish Middle Ages. According to the medieval tradition, he was the Bishop of Uppsala (Sweden), and one of the leaders of a Swedish expedition to Finland around 1155, during which he was murdered. The oldest text concerning St. Henry is his legend written in Latin by the end of the 13th century at the very latest.

[5] For identification of the sources as well as a modern edition of the legend see [10].

16th centuries (Fig. 1). In the relatively short text there are nearly one thousand places where the versions differ from each other. Since the multitude of possible stemmata rises easily to astronomic numbers, 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 the aid of computer science. In particular, algorithms 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 than before. 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 2 we present a criterion for stemmatic analysis that is based on compression of the manuscripts. The intuitive idea behind compression-based approaches is that if a text can be significantly compressed, then the compression algorithm has found regularities which can be further exploited in an analysis such as ours. We then outline in Sec. 3 an algorithm that searches in the space of tree-shaped stemmata and chooses the one that minimizes the criterion. The method is illustrated on a simple example in Sec. 4, where we also present our main experiment using all 52 known 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. 5.

2 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 the 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 [23] in Robinson's Textual Criticism challenge [19]. For further applications of maximum parsimony and related method, see [11, 15, 22, 25] and the 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 [18] — 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 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, describing a deletion of a word or a change of word order requires less information than 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 [13, 16], 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, is fundamentally uncomputable. In the spirit of a number of earlier authors [1, 2, 3, 5, 9, 17, 24], we approximate Kolmogorov complexity by using a compression program. Currently, we use `gzip` based on the LZ77 [26] 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[6]. One of the advantages of using a string compression method that operates directly on the text is that only minimal preprocessing (see below) is required, contrary to most of the methods mentioned above. A simple example illustrating these concepts is given below in Sec. 4.

In addition to the MDL justification, our method can be seen as (an approximation of) maximum likelihood, another commonly used criterion in phylogeny that has good properties in terms of theoretical (consistency) guarantees and empirical performance [6]. 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 [21]. Code-lengths have an interpretation in terms of likelihoods: 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 corresponding to the text variants, $E \subset V \times V$ is a set of edges. We require that the graph is a connected bifurcating tree, meaning 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 the 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

$$C(\vec{G}) = \sum_{v \in V} C(v \mid \mathrm{Pa}(v)) \quad (1)$$

$$= \sum_{v \in V} C(\mathrm{Pa}(v), v) - C(\mathrm{Pa}(v)), \quad (2)$$

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 be easily verified that for bifurcating trees, the above can be rewritten as

$$C(G) \approx \sum_{(v,w) \in E} C(v, w) - 2 \sum_{v \in V_I} C(v), \quad (3)$$

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

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

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 (in the experiment reported in Sec. 4, we used 11). In order for the segments to cover the same part of the text, the variants need to be word-by-word aligned, which can usually be achieved with relatively minor effort. 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[7]. Thus, we use the following modified cost function

$$C'(\vec{G}) = \sum_{v \in V} \sum_{i=1}^{n} C'(v_i \mid \mathrm{Pa}_i(v)), \tag{4}$$

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 (3) and hence, is practically independent of the choice of the root.

3 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 (4) 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 (4). 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 to 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 nodes is an instance of dynamic programming and called 'the Sankoff algorithm' [6] or 'Felsenstein's algorithm' [20]. As Siepel and Haussler [20] note, it is in fact an instance of a 'message-passing' or 'elimination' algorithm in graphical models (see also [8]). 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 [6])

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

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 $\mathrm{cost}_i(j)$ 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 $\mathrm{cost}_{\mathrm{root}}(j) + C'(x^j)$. 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. (3)).

There still remains the problem of finding the tree structure, which together with corresponding optimal contents of the hidden nodes minimizes criterion (4). 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 exponentially large (see [6]). 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 [12] that starts with an arbitrary tree and iteratively tries to improve it by small random modification, such as exchanging the places of two subtrees[8]. 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

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

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; hence the name 'simulated annealing'. In our main experiment, reported in the next section, we performed several runs of up to

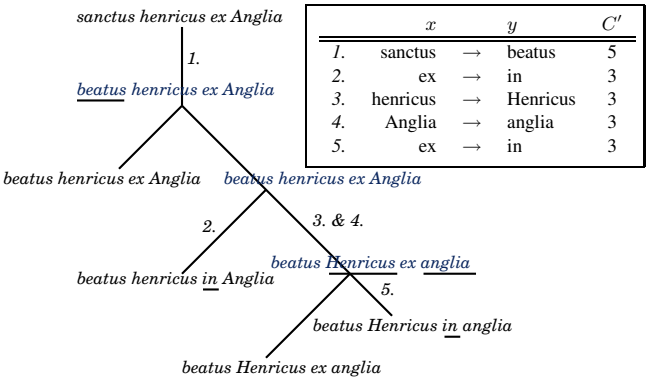
[7] Howe *et al.* [11] use as an example the words *kirk* and *church* in 15th century English whose variation mainly reflects local dialect.

[8] 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.

2,500,000 iterations, which we found to be sufficient in our setting. The best tree of all the runs was then retained as the final outcome[9].

4 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.



	x		y	C'
1.	sanctus	→	beatus	5
2.	ex	→	in	3
3.	henricus	→	Henricus	3
4.	Anglia	→	anglia	3
5.	ex	→	in	3

Figure 2. An example tree obtained with the compression-based method for the five strings at the tips of the branches. Changes are underlined and numbered. Costs of changes are listed in the box. Best reconstructions at interior nodes are shown at the branching points.

As our main experiment, we analyzed all the known 52 variants of the legend of St. Henry. The variants contained 23–942 words each. The best (wrt. the information cost) tree found is shown in Fig. 3. By comparing the tree with earlier results [10], 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[10]. Other pairs of variants appearing in the tree that are believed to be directly related are JG–B, O–P, NR2–JB, LT–E, AJ–D, and MN–Y. In addition, the subtree including the ten nodes starting at (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, and suggests many groups that are well in line with the evidence but have been previously unrecognized.

[9] We also used bootstrapping to evaluate the confidence in the result but due to space restrictions we present only results pertaining to the single best tree.

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

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 [8]). In this case we should also account for possible forms (words or segments) not occurring in any of the observed variants. However, if this could be done, one could handle arbitrary graph structures, although this may be computationally demanding.
6. The search space is huge and the algorithm only finds a local optimum whose quality cannot be guaranteed. Bootstrapping [14] can be used to identify which parts of the tree are uncertain due to problems in search (as well as due to lack of evidence).

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 [4], Phylip [7], and PAUP [23]. We are also planning to make the implementation publicly available. 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.

5 CONCLUSIONS

We proposed a new compression-based criterion, and an associated algorithm for computer-based 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 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.

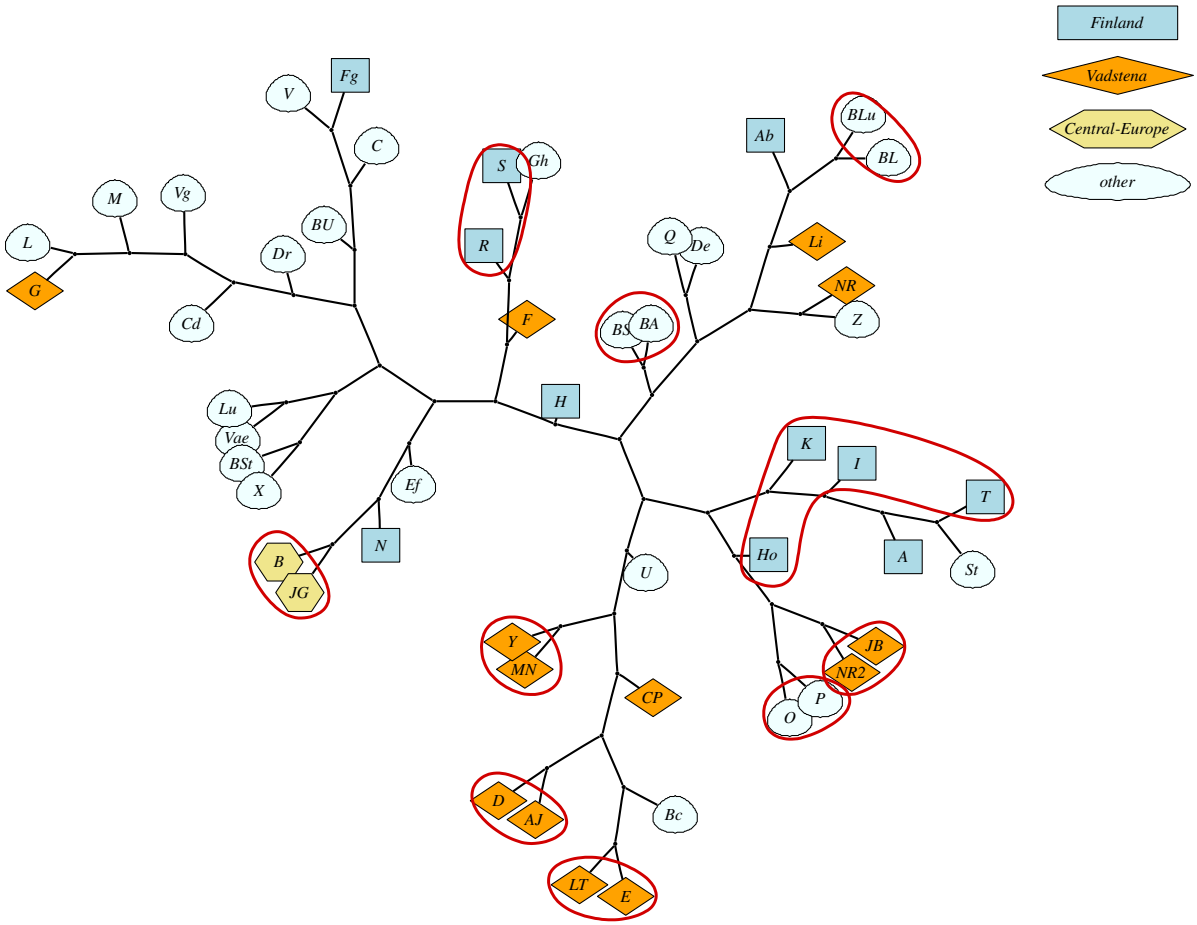


Figure 3. Best tree found. Most probable place of origin according to [10] indicated by color and shape — Finland (blue rectangle): Ab,K,Ho,I,T,A,R,S,H,N,Fg; Vadstena monastery in Sweden (red diamond): AJ,D,CP,E,LT,MN,Y,JB,NR,NR2,Li,F,G; Central Europe (yellow hexagon): JG,B; other, mostly Sweden (white oval). Some groups supported by earlier work are circled in red.

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. We are currently carrying out controlled experiments with artificial data with known 'ground-truth' solution to which the results can be compared. Outside historical and biological applications, analysis of computer viruses is an interesting future research topic.

ACKNOWLEDGEMENTS

This work has significantly benefited from discussions with Tommi Mononen and Kimmo Valtonen at HIIT, and Prof. Paul Vitányi and Rudi Cilibrasi at CWI. This work was supported in part by IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views.

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, M. Li, and B. Ma, 'Chain letters and evolutionary histories', *Scientific American*, 76–81, (November 2003).

[3] X. Chen, S. Kwong, and M. Li, 'A compression algorithm for DNA sequences and its applications in genome comparison', in *Genome Informatics*, eds., K. Asai, S. Miyano, and T. Takagi, Tokyo, (1999). Universal Academy Press.

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

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

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

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

[8] 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).

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

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

[11] 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).
[12] S. Kirkpatrick, C.D. Gelatt Jr., and M.P. Vecchi, 'Optimization by simulated annealing', *Science*, **220**(4598), 671–680, (1983).
[13] A.N. Kolmogorov, 'Three approaches to the quantitative definition of information', *Problems in Information Transmission*, **1**(1), 1–7, (1965).
[14] H.R. Künsch, 'The jackknife and the bootstrap for general stationary observations', *Annals of Statistics*, **17**(3), 1217–1241, (1989).
[15] A.-C. Lantin, P. V. Baret, and C. Macé, 'Phylogenetic analysis of Gregory of Nazianzus' Homily 27', in *7èmes Journées Internationales d'Analyse statistique des Données Textuelles*, eds., G. Purnelle, C. Fairon, and A. Dister, pp. 700–707, Louvain-la-Neuve, (2004).
[16] M. Li and P.M.B. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications, 2nd. Ed.*, Springer-Verlag, New York, 1997.
[17] D. Loewenstern, H. Hirsh, P. Yianilos, and M. Noordewier, 'DNA sequence classification using compression-based induction', Technical Report 95–04, DIMACS, (1995).
[18] J. Rissanen, 'Modeling by shortest data description', *Automatica*, **14**, 465–471, (1978).
[19] P. Robinson and R.J. O'Hara, 'Report on the textual criticism challenge 1991', *Bryn Mawr Classical Review*, **3**(4), 331–337, (1992).
[20] A. Siepel and D. Haussler, 'Phylogenetic estimation of context-dependent substitution rates by maximum likelihood', *Molecular Biology and Evolution*, **21**(3), 468–488, (2004).
[21] M. Spencer and C.J. Howe, 'How accurate were scribes? A mathematical model', *Literary and Linguistic Computing*, **17**(3), 311–322, (2002).
[22] 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).
[23] D.L. Swofford. PAUP*: Phylogenetic analysis using parsimony (*and other methods). version 4., 2003.
[24] 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).
[25] E. Wattel and M.P. van Mulken, 'Weighted formal support of a pedigree', in *Studies in Stemmatology*, eds., P. van Reenen and M.P. van Mulken, 135–169, Benjamins Publishing, Amsterdam, (1996).
[26] J. Ziv and A. Lempel, 'A universal algorithm for sequential data compression', *IEEE Transactions on Information Theory*, **23**(3), 337–343, (1977).