

# MDL hierarchical clustering with incomplete data

Po-Hsiang Lai

Electrical and Systems Engineering  
Washington University in Saint Louis  
Email: pl1@wustl.edu

Joseph A. O’Sullivan

Electrical and Systems Engineering  
Washington University in Saint Louis  
Email: jao@wustl.edu

**Abstract**—The goal of stemmatology is to reconstruct a family tree of different variants of a text resulting from imperfect copying, which is a crucial part of textual criticism. In reality, historians often have incomplete data because some variants are not yet discovered and there are missing portions in available variants due to physical damage. Stemmatology is similar to molecular phylogenetics where biologists aim to reconstruct the evolutionary history of species based on genetic or protein sequences. Adoption of phylogenetics methods has led to encouraging results in automatic stemmatology. We discuss and demonstrate the potential application of minimum description length (MDL) concepts to stemmatology. Our method is applied to a realistic dataset and outperforms major existing methods.

## I. INTRODUCTION

Before printing technology was widespread, text documents had to be copied by hand, mostly with errors. Thus, despite many documents originating from a common original text, they differ from one another. For those variants that survived and were discovered, historians are interested in knowing the relations among them, in particular, the family tree of the copying history. The research of finding such a family tree based on surviving variants is called stemmatology, and a proposed tree is called a stemma. A stemma is ideally a rooted tree where a child node is copied from its ancestor node in the tree. An accurate stemma with geographical, and temporal if available, information of variants, may provide important historical evidence related to the spread and interaction of variants with local cultures.

There are a number of mechanisms which lead to differences in variants. During the Middle Ages, Latin was no longer an actively spoken or written language. However, many texts were still copied in Latin; the copyists might understand a part of the text. This results in a large amount of unintentional copying error as well as intentional changes. Also for an original text being copied for centuries, the errors accumulate from one copy to another. These have resulted in large differences among surviving variants. Also, to construct a stemma, a number of variants must be considered simultaneously. The number of possible stemmata grows enormously with the number of variants: for example, there are  $1.4 \times 10^9$  stemmata for 30 variants [1]. Hence, beyond traditional manual approaches, computer aided stemmatology methods are needed.

One can quickly notice that the problem of stemmatology is closely related to phylogenetics. The copying process with error is similar to genetic mutations during the evolution process. Also in both cases, there are missing variants. In

biology, there may be no genetic data from extinct species. For both cases, variants whose word orders or genetic sequences are similar to each other are considered to be close in the resulting tree. Many automatic stemmatology methods are inspired by phylogenetic methods, and have been improved since the work of Robinson and O’Hara [2]. These methods have produced encouraging results as they have been applied and evaluated on small datasets where historians have strong confidence in historical relation among variants. For these datasets, there is a consensus stemma based on many forms of evidence.

Despite those successful automatic stemmatology results, several challenges remain. In particular, early test datasets are relatively small and ideal in that there are few missing variants, and most available variants have few missing portions. However as mentioned before, it is known that historical variants have missing portions due to physical damage. This poses additional challenges compared to phylogenetics, where in most cases, full gene or protein sequences are available. On the other hand, it is reasonable for phylogenetics to construct a bifurcation tree with all variants as leaves since there rarely is an occasion where more than two species mutated and evolved from an ancestor at exactly the same time, and the surviving species should indeed be the result of the latest mutations. This is not the case in stemmatology, where several copyists can copy from an identical source and surviving variants need not be the latest copies. Other issues such as contamination where a variant is copied based on two or more sources are also unique in stemmatology. In the review by Roos and Heikkilä [3], 13 major algorithms are evaluated on three artificially generated datasets with known true stemmata. The datasets are generated by subjects copying texts but not real historical data. Notably one of them, the *Heinrichi* dataset, is a much more realistic dataset where nearly half of the variants are missing, and available variants have large missing portions. For more detail see Section IV. Even though the best performing method on the *Heinrichi* dataset obtains good accuracy slightly lower than results from simpler datasets, surprising failures of several promising methods, such as `CompLearn`, indicate that more should be done to address the issue of incomplete data [4].

In this paper, MDL concepts [5], [6], [7], [8] with applications to stemmatology are discussed. MDL is data driven and need not assume a true distribution or model of the data. Instead of assuming the resulting stemma is a bifurcation tree and all variants are the leaves, we infer the stemma as one

which minimizes the number of bits it takes to describe all variants based on a given encoding method. This idea is a direct application of code length based MDL clustering [9]. In the code, a variant can either be encoded by itself, or encoded using another variant as an exemplar, i.e. in the tree as ancestor. The resulting stemma is directly determined by data. Due to missing portions, the code length between two variants has to be estimated and adjusted according to what the missing locations are. Similarly in Cilibrasi and Vitányi [10], code length is an ingredient to obtain a better distance measure, the Normalized Compression Distance (NCD), between data points, while the resulting clustering does not have a direct code length interpretation. NCD is used in `CompLearn` which surprisingly failed the *Heinrichi* dataset. We discuss the possible reasons why the NCD based method fails [4] and why our methods and the RHM algorithm developed by Roos, *et al.* [11], [3], work on the *Heinrichi* dataset.

This paper is organized as follows. Basic notation and terminology are introduced in section II. In section III, computational challenges in stemmatology are discussed, as well as our reasons for selecting a particular angle to approach them. The datasets are introduced in section IV. The concepts of MDL based clustering and NCD based clustering are discussed in section V with their applications in stemmatology. We also discuss the performances of NCD based `CompLearn` and the RHM algorithm. In section VI, we present a preliminary approach and simulation results. In section VII we propose several possible future directions to further improve the result.

## II. NOTATION AND TERMINOLOGY

We denote a variant as  $x_j = (x_1^j, \dots, x_n^j)$ , where  $j$  is the variant index, and  $n$  is the supposed number of words in a variant. For each location  $i$ ,  $x_i^j, i = 1, \dots, n$  is either a word or “?” denoting that the word is missing. The set of all  $N$  available variants is  $S$ . The set of all distinct words appearing in variants is  $X$ , and  $m$  is the total number of words in  $X$ . We assume that the all variants are aligned and do not focus our attention on alignment methods. A stemma of a set of variants is a connected graph  $G = (V, E)$  such that  $S$  is a subset of  $V$ . Clearly,  $V$  may contain auxiliary nodes, as it is known that there are missing variants that might be inferred. Note that a stemma need not be a tree due to contamination where a variant is copied from multiple sources.

To compare between stemmata, the *average sign similarity* is introduced by Roos and Heikkilä [3]. For a given undirected graph  $G$ , the simple path length between two nodes  $A$  and  $B$  is defined as the smallest number of edges needed to connect  $A$  and  $B$  on  $G$ . On the true graph, it is denoted as  $d(A, B)$ , whereas on the inferred graph it is denoted as  $d'(A, B)$ . For any three nodes  $A, B$ , and  $C$ , the sign agreement index is defined as

$$u(B, C|A) = 1 - \frac{1}{2} |\text{sgn}(d(A, B) - d(A, C)) - \text{sgn}(d'(A, B) - d'(A, C))|, \quad (1)$$

where  $|\cdot|$  is the absolute value. The main idea is to check if  $B$  and  $C$  have the same ordering related to a reference  $A$  on

a proposed stemma and the true stemma. It is equal to 1 if the ordering is matched,  $1/2$  if one and only one of them is zero, and 0 if the ordering is mismatched. The *average sign distance* between two stemmata  $G$  and  $H$  is defined as

$$D(G, H) = \sum_{x_i \neq x_j \neq x_k} u(x_j, x_k|x_i)/6. \quad (2)$$

The sum is over all triples of variants. Dividing by 6, we discount triples that are equivalent after permutation. Let  $T$  denote the true stemmata, the score of an inferred structure  $G$  is defined as  $D(G, T)/D(T, T)$ .

## III. COMPUTATIONAL CHALLENGES IN HIERARCHICAL CLUSTERING WITH INCOMPLETE DATA

Inferring structure among data points in the presence of missing data is tied closely to a set of graphical optimization problems collectively called the Steiner Tree problem. In the Steiner tree problem, a graph  $G = (V, E)$  and a subset  $S$  of  $V$  are given. The goal is to find a tree  $G'$  that connects all nodes in  $S$  and minimizes the total edge weights in  $G'$ . In stemmatology and phylogenetics,  $S$  is then the set of variants or genetic sequences of interests, and  $V$  is then the set of all possible variants or genetic sequences that are relevant to the problem at hand. The edge weight is a distance or similarity measure we pick. In general, the Steiner tree problem is  $\mathcal{NP}$  hard, and it is even  $\mathcal{NP}$  to have a close approximation. Under the case with missing portions in available variants, in the worst cases, even the optimal imputation and structural inference among only the available variants is a Steiner Tree problem.

Rather than adopting to our problem a general algorithm for the Steiner tree problem, we develop a novel approach based on the specific properties of the datasets in stemmatology. Our goal is to learn what concepts should yield algorithms that perform well. The current best performing algorithm RHM developed by Roos, *et al.*, is in fact closely related to a Steiner tree algorithm. We give an MDL interpretation of why RHM succeeds.

## IV. DATASETS

The *Heinrichi* dataset and the *Parzival* dataset are used to evaluate algorithm performances in the Computer-Assisted Stemmatology Challenge [3]. The *Heinrichi* dataset consists of an original text, a 17th century late medieval Finnish folktale *Piispa Henrikin Surmavirsi*, written in old Finnish. 17 copyists participated to produce 67 text variants with contamination. The copyists are mostly Finnish but can only understand some ancient words, which resembles the situation in real stemmatology problems. In simulation, large portions of available variants are deleted on purpose, and only 37 variants are available. Thus it is similar to real world stemmatology with the physical damage and variants uncovered. Each variant has around 1200 words with an average of 300 missing words. *Heinrichi* is currently the most realistic data set with a large amount of incomplete data.

On the other hand, the *Parzival* [3] dataset is smaller consisting of 21 variants of the German poem *Parzival* by Matthew Spencer and Heather F. Windram. Only 5 out of the 21 variant are missing to the algorithm and no missing portions except those generated by copying error. This dataset is mostly for validation that any algorithm should produce reasonable performance on it, and it is easy to analyze resulting on this dataset.

## V. MDL, COMPRESSION AND CLUSTERING

Clustering is closely related to the problem of structural inference. In clustering, data points are similar to one another should be assigned to the same group, whereas in structural inference, data points that are similar to one another should be assigned closely on the tree. The relation is particularly clear when one considers similarity based clustering, where the input to the clustering algorithm is an  $N$  by  $N$  matrix with each  $i, j$ -th element being the discrepancy from the  $i$ -th element to the  $j$ -th element. Note that this is different from model based clustering which is another popular class of cluster methods using estimated models, usually densities such as Gaussian Mixture Models (GMM), to partition the data. It is intuitive to utilize clustering ideas in structural inference, and a similarity measure between data points has to be chosen properly. Recently, several similarity measures and similarity based clustering methods using information theoretic ideas have been proposed [10], [9]. The fundamental elements across these works is to measure similarity directly in terms of bits [9] (MDL based clustering), or through a function whose inputs have units in bits [10] (normalized compression distance based clustering). Algorithms closely related to these two types of ideas were developed and applied to stemmatology (see the review by Roos, *et al.*) [11]. Here we briefly review the two information theoretic ideas for distanced based clustering and their related stemmatology methods. We follow with a discussion on why RHM works and `CompLearn` fails due to incomplete data.

### A. MDL based clustering

The idea of the minimum description length principle (MDL) is to select a model that requires the least number of bits to generate the data of interest. The total count of bits includes both the bits used to describe the model and the bits to describe the data given the model. Hence it naturally balances model complexity and data fitness. Likewise, we can think of clustering as an idea of grouping data points so that given the grouping, it takes the least number of bits to describe the data [9]. Ideally, one would like to compute the total bits needed to describe the data given any data partition and then choose the most efficient one as the clustering result. However this is computationally expensive as the number of different partitions is exponential in the number of data points. Instead, for each data point we can focus on two ways of encoding. One way is to encode a data point  $x_i$  by itself using a chosen compression method, the resulting code length being denoted as  $L(x_i)$ . The other way is to encode a data point  $x_i$  given

another data point  $x_j$ , the resulting code length being denoted as  $L(x_i|x_j)$ . To efficiently describe all the data points given these two ways of encoding, one has to find the right ordering of which data points should be encoded and followed by which other data points. The objection function is then

$$L(x_1, \dots, x_n|\mathbf{t}) + L(\mathbf{t}) = \sum_{x_i:i \neq t_i} L(x_i|x_{s_i}) + \sum_{x_i:i=t_i} L(x_i) + L(\mathbf{t}), \quad (3)$$

where  $\mathbf{t} = (t_1, \dots, t_n)$  is a vector of indexes where  $t_i$  denotes the index of the ancestor of  $x_i$ . We can think of a graph  $G = (V, E)$ , where  $V$  consists of all  $N$  data points and a root node, and  $E$  is the set of directed edges. Each directed edge is assigned a weight equal to its code length. An edge pointing from the  $j$ -th data point to the  $i$ -th data point has code length  $L(x_i|x_j)$ . An edge from the root node to data point  $i$  has code length  $L(x_i)$ . Thus, one seeks the subgraph which starts from the root and passes through all vertices with the minimum total edge length. Clearly there should be no cycle that the subgraph is in fact a tree. It is well known as the minimum arborescence tree for directed graphs, and as the minimum spanning tree (MST) for undirected graphs. For clustering, the number of children of the root node is considered to be the number of clusters, and data points with the same ancestor are considered to be in the same cluster [9].

In the RHM algorithm, which is independently developed by Roos, *et al.* [11], [3], the code length of  $L(x_i|x_j)$  is computed as

$$L(x_i|x_j) = L(x_i, x_j) - L(x_i). \quad (4)$$

This is based on Kolmogorov complexity and coding theory [12], [13], [8] that  $L(x_i, x_j) \leq L(x_i) + L(x_j)$ , i.e. joint compression of two sequences takes less bits than the total bits for compression individually. Hence given that  $x_i$  is available, one needs only  $L(x_i, x_j) - L(x_i)$  bits instead of  $L(x_j)$  bits. The compression algorithm used in RHM is `gzip`, which uses LZ77 and Huffman coding. Due to the existence of missing variants, RHM produces a stemma which is a bifurcation tree with all variants being the leaves, similar to assumptions made in phylogenetics. The internal nodes are auxiliary variants created such that they minimize the total bits in the tree. The process is done iteratively so that for a fixed stemma, it chooses optimal auxiliary variants, and then computes the bifurcation tree based on fixed auxiliary variants. Prior to our work, the RHM algorithm was the best performing algorithm for the *Heinrichi* dataset, the most complicated stemmatology dataset [3].

### B. Normalized Compression Distance based clustering

The Normalized Compression Distance (NCD) was first proposed by Cilibrasi and Vatányi [10] for clustering. The main idea is to have a theoretically sound distance measure for clustering methods. It starts with Kolmogorov complexity and shows that a distance measure can be derived through proper normalization. However, the Kolmogorov complexity is in general not computable in the sense that no general algorithm can determine if a given code of data has the

shortest code length possible. However, it is indeed possible to use an actual compression algorithm such as **gzip** and to use the resulting code length to derive an analogous distance measure though almost the same arguments and normalization used for Kolmogorov complexity based distance. This distance measure depends on the use of a compression algorithm and normalization, hence the name Normalized Compression Distance. Computationally, the NCD is defined as

$$e_L(A, B) = \frac{L(A, B) - \min\{L(A), L(B)\}}{\max\{L(A), L(B)\}}. \quad (5)$$

Give a set of data, one can then compute pairwise NCD, store in a matrix  $M$ , and feed into a distance based clustering algorithm. For stemmatology, one inputs  $M$  to an algorithm which generates a bifurcation tree as the inferred stemma. In general, edge lengths are put in the resulting bifurcation tree, such that pairwise distance of available variants are preserved in the tree.

### C. RHM versus NCD based stemmatology

The NCD based method `COMPLEARN` seems to fail on the *Heinrichi* dataset. In the *Heinrichi* dataset, about half of the variants are missing, which is taken into account by both RHM and `COMPLEARN` through putting all available variants as leaves and inferring the locations of missing variants in a bifurcation tree. An other feature of the *Heinrichi* dataset is the large size of missing portions in available variants: on average of  $\frac{1}{4}$  of the words are missing up to  $\frac{3}{4}$ . Due to missing portions, it may take a lot of bits to encode a variant  $x_i$  given another closely related variant  $x_j$  simply because a large portion of  $x_i$  is the missing portion of  $x_j$ . This issue seems to be taken care of by the normalization part of NCD and not in the RHM algorithm. However, as discussed in [4], the normalization is biased in that variants that have a large common missing portion are considered closer. To see this, observe that in the numerator, it is a max operation of two code lengths while in the denominator, there is a min operation of the same two code length. Thus if two variants  $A$  and  $B$  have similar length and missing portions,  $L(A, B)$ ,  $\min\{L(A), L(B)\}$ , and  $\max\{L(A), L(B)\}$  would be similar, which results in an NCD close to unity. However, when the length of  $A$  is much larger than  $B$ ,  $L(A, B)$  would be similar to  $\max\{L(A), L(B)\} = L(A)$  but both much larger than  $\min\{L(A), L(B)\} = L(B)$ . However, this does not explain the success of the RHM algorithm, which does not consider normalization at all.

Even though both RHM and `CompLearn` generate bifurcation trees, the mechanisms are quite different. The RHM algorithm explicitly generates auxiliary variants whereas `CompLearn` preserves distances on the graph without estimating additional nodes. This is in fact a possible reason why RHM works. Consider two variants  $A = (10010???)$  and  $B = (10010111?)$ . They may have a large compression distance. But if we add a variant  $C = (100101110)$  as their parent, clearly it is easy for  $C$  to encode both  $A$  and  $B$  as in compression, missing portions require nearly no bits to encode,

and for the common available portions,  $B$  and  $C$  are indeed identical. Thus in RHM, iteratively changing the stemma and updating auxiliary variants may help join variants that are in fact similar to one another but have different portions missing. This is not possible for NCD based methods, where information about the variants are represented by a distance matrix.

Also, this points out an important idea that MDL is applicable to guide where and how many auxiliary nodes are needed. Thus, it frees the structure from being a bifurcation tree which might not be the best fit for stemmatology problems. Although the resulting problem is still a Steiner Tree problem, it could provide guidance of local update. Consider three variants  $A, B, C$  with  $A$  as the parent, to determine if there is a need to add another variant  $D$  one can compute whether

$$L(D|A) + L(B|D) + L(C|D) < L(B|A) + L(C|A). \quad (6)$$

If so, it implies that adding a new variant is in fact reducing the code length so that in an MDL sense, it is reasonable to do so.

## VI. SIMULATION RESULTS

From the previous section, we can observe that to succeed in hierarchical clustering with incomplete data, a key factor is that the algorithm needs to be able to get around possible bias due to missing portions in available data. On the other hand, Steiner tree type algorithms that iteratively generate auxiliary variants and update inferred structure seem promising. Here we focus on a simple approach to deal with the missing portion issue, and leave the integration of inferring missing variants using Steiner Tree algorithms with MDL interpretation as future work. Without adding auxiliary nodes, it is harder to directly use the code length between two variants. However, it can be estimated by a very simple method, the normalized Hamming distance. For a simple encoder, it searches the locations where two variants are different, and encodes the difference with a uniform prior on the locations and words of the difference. Thus the code length is closely proportional to the number of differences given two variants. With missing portions, one can still have a rough estimate about what the code length should be if two variants were complete. We introduce the normalized Hamming distance. For two variants  $i$  and  $j$ , the normalized Hamming distance is  $\frac{k_{ij}}{n_{ij}}$ , where  $k_{ij}$  is the number of differences in the portions that are available in both variants, and  $n_{ij}$  is the total length of the overlapping portions. Assuming that the statistics of copying error are roughly the same before and after damage, the number of differences between variants  $i$  and  $j$  if they were complete is approximately  $n \frac{k_{ij}}{n_{ij}}$ . This assumption makes intuitive sense that the process of physical damage should be independent of the copying errors. Note that the minimum spanning tree of a graph  $G$  is invariant to a constant scaling of the edge weights. Thus, since the code length is proportional to the number of differences, which is proportional to the normalized Hamming distance, we can use the normalized Hamming distance of all

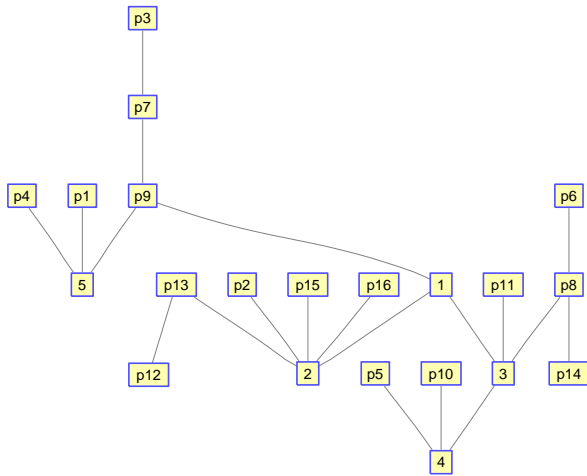


Fig. 1. The true stemma of the *Parzival* dataset. The nodes labeled with pure numbers are missing variants that are not available to the algorithm

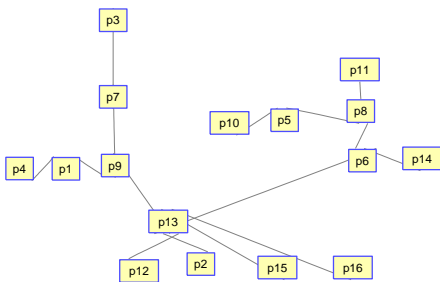


Fig. 2. The stemma result from the minimum spanning tree based on the normalized Hamming distance for the *Parzival* dataset. Note that in the true stemma, there are five variants not in the inferred stemma. This results in several errors in the sign similarity measure. For example variant 8 is directly connect to variant 5 and variant 6 in the inferred stemma, while there are actually two missing variants between variant 8 and 5. On the other hand, in the true stemma if we view two variants connected though unavailable variants as directly connected, the inferred structure is actually close to the true structure.

pairwise variants as the input to the minimum spanning tree algorithm to infer a stemma with MDL interpretations.

Using the normalized Hamming distance, we achieve 79.0% accuracy on the *Heinrichi* dataset, which is better than all 13 algorithms reviewed in [3]. The result is 78.5% on *Parzival* datasets, which is about the average performance among the 13 algorithms while only slightly lower the performance of the RHM algorithm 79.9%. The resulting tree is shown in Figure 2. The detailed results for the *Heinrichi* dataset will appear in subsequent publications.

## VII. CONCLUSIONS

We discuss the use of MDL concepts in hierarchical clustering with incomplete data, and we use stemmatology

problems as example applications. We give insights that the successful RHM algorithms have nice MDL interpretations for dealing with both missing portions of available variants and missing variants. We introduce the use of normalized Hamming distance and minimum spanning tree idea to handle missing portions in variants with direct MDL interpretations, and obtain very encouraging good results on realistic stemmatology datasets. The idea of MDL and Steiner Tree have a great potential in developing new algorithms for hierarchical clustering with incomplete data. The results from normalized Hamming distance can be used as a initialization step for more complicated algorithms.

## ACKNOWLEDGMENT

This work was supported in part by System Dynamics International. The stemmatology problem was brought to our attention by Teemu Roos.

## REFERENCES

- [1] J. Felsenstein, *Inferring Phylogenies*. Sinauer Associates, Inc., 2004.
- [2] P. Robinson and R. J. O'Hara, "Report on the textual criticism challenge 1991," *Bryn Mawr Classical Review*, vol. 3, no. 4, pp. 331–337, 1992.
- [3] T. Roos and T. Heikkilä, "Evaluating methods for computer-assisted stemmatology using artificial benchmark data sets," *Literary and Linguistic Computing*, vol. 24, no. 4, pp. 417–433, 2009.
- [4] T. Merivuori and T. Roos, "Some observations on the applicability of normalized compression distance to stemmatology," ser. Proceedings of 2nd Workshop on Information Theoretic Methods in Science and Engineering, 2009.
- [5] J. Rissanen, "Fisher information and stochastic complexity," *IEEE Trans. Info. Theory*, vol. 42, pp. 40–47, 1996.
- [6] A. Barron, J. Rissanen, and B. Yu, "The minimum description length principle in coding and modeling," *IEEE Trans. Info. Theory*, vol. 44, pp. 2473–2760, 1998.
- [7] P. D. Grünwald, *The Minimum Description Length Principle*. MIT press, 2007.
- [8] J. Rissanen, *Information and Complexity in Statistical Modeling*. Springer, 2007.
- [9] P.-H. Lai, J. A. O'Sullivan, and R. Pless, "Minimum description length and clustering with exemplars," ser. Proceedings of 2009 IEEE International Symposium on Information Theory, Seoul, Korea, 2009.
- [10] R. Cilibrasi and P. M. B. Vitányi, "Clustering by compression," *IEEE Trans. Info. Theory*, vol. 51, pp. 1523–1524, 2005.
- [11] T. Roos, T. Heikkilä, R. Cilibrasi, and P. Myllymäki, "Compression based stemmatology: a study of the legend of St. Henry of Finland," in *Helsinki Institute for Information Technology technical report 2005-3*.
- [12] M. Li and P. Vitányi, *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2008.
- [13] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. New York, NY.: John Wiley and Sons, Inc., 1991.