AncestryAI: A Tool for Exploring Computationally Inferred Family Trees

Eric Malmi
Aalto University
Espoo, Finland
eric.malmi@aalto.fi

Marko Rasa*
Verto Analytics
Espoo, Finland
marko.rasa@vertoanalytics.com

Aristides Gionis
Aalto University
Espoo, Finland
aristides.gionis@aalto.fi

ABSTRACT

Many people are excited to discover their ancestors and thus decide to take up genealogy. However, the process of finding the ancestors is often very laborious since it involves comparing a large number of historical birth records and trying to manually match the people mentioned in them. We have developed AncestryAI, an open-source tool for automatically linking historical records and exploring the resulting family trees. We introduce a record-linkage method for computing the probabilities of the candidate matches, which allows the users to either directly identify the next ancestor or narrow down the search. We also propose an efficient layout algorithm for drawing and navigating genealogical graphs. The tool is additionally used to crowdsource training and evaluation data so as to improve the matching algorithm. Our objective is to build a large genealogical graph, which could be used to resolve various interesting questions in the areas of computational social science, genetics, and evolutionary studies. The tool is openly available at: `http://ancestryai.cs.hut.fi/`.

1. INTRODUCTION

Many people are intrigued by their roots. This is reflected in the popularity of various commercial genealogy websites, such as *Ancestry.com* and *MyHeritage*, and TV series, such as *Who Do You Think You Are?* In Finland, the publicly available parish registers are particularly extensive and a dataset of over 10 million birth, marriage, burial, and migration records, digitized and transcribed by volunteers, is openly accessible [12]. In this paper, we introduce a probabilistic method for matching and linking these records. We develop an open web-based tool, called AncestryAI, which is meant for exploring the family trees (genealogical graphs) that are computationally inferred from the data.

The main motivations for creating AncestryAI are the following:

1. Provide an open-source tool,[1] which makes genealogical research faster and possibly more accurate.
2. Collect ground-truth data to improve computational family-tree inference.
3. Create a large-scale genealogical graph for computational social science studies.

Let us now discuss in more detail these three points.

First, computational matching of population records makes genealogical research faster: in cases with no ambiguity the user can directly obtain the matching records, while, when the matches are ambiguous, e.g., due to duplicate names, our probabilistic inference algorithm helps to narrow down the search by listing the most probable matches ranked by their probabilities. In some cases, AncestryAI can also improve the quality of the resulting family tree, as the algorithm can analyze a very large dataset as a whole, and thus, it can identify ancestors who have moved from other cities, whereas the researcher alone might need to limit the search to nearby areas to make the task more feasible.

Second, AncestryAI has a feature that allows users to annotate correctly and incorrectly matched individuals. This allow us to collect more training data so as to improve the estimation of the matching probabilities and evaluate the algorithm more reliably.

Third, by collecting the user annotations and improving the algorithm, we aim to create a large genealogical graph with up to millions of nodes, which covers three centuries and different areas in Finland. Even though such a graph will never be error free, it can help us discover and quantify large-scale societal phenomena taking place over long periods of time. It would be interesting to study, for example, social mobility as observed through marriage patterns, or migration patterns over time [14]. Unlike the data collected from social media, which is the main source of computational social science studies nowadays [7], the genealogical graph could help us study phenomena spanning several generations. Other interesting applications include genetic [11] and evolutionary studies [10].

In addition to providing a novel open-source tool for the genealogy community, in this work we make the following methodological contributions:

*The majority of this work was completed while the author was a research assistant in Aalto University.

[1]The source code of AncestryAI is available at: `https://github.com/ekQ/ancestryai`

WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054728

1. We derive a probabilistic method for matching historical records that allows, not only inference of the most probable ancestors, but also quantifying the uncertainty of the matches.

2. We introduce an efficient layout algorithm for drawing family trees.

2. DATA

The publicly available parish registers in Finland are particularly extensive, dating from the 1600s to the late 1800s (more recent data is not publicly available due to current legislation). The "HisKi" project, an effort started in the 1980's, aims to digitize parts of the Finnish parish registers. These registers can be considered an early population register, which was kept by the Evangelical Lutheran Church, a national church in Finland. The data contains about 5 million records of births and a total of 5 million records of deaths, marriages and migration. The coverage of the records was originally close to full, but the digitized material covers only parts of the complete dataset (some material is not digitized yet, and some is lost). The Genealogical Society of Finland maintains a web service where users can query the records [12].

Each birth record (more precisely, baptism record) typically contains the name, birth place, and birth date of the child in addition to the names of the parents. All of these attributes are useful for inferring the most likely birth records of the parents. Finding the birth records of the parents is, however, *not a trivial problem* due to spelling variations, duplicate names, and missing records.

Additionally, we have obtained a family tree containing 64 208 people constructed by an individual genealogical researcher from Finland. It is used as ground-truth data for estimating the likelihoods required by the model as discussed later in Section 4.3.

Currently, AncestryAI uses only Finnish data but it could easily be used for building and exploring family trees in other countries.

3. MAIN FEATURES

AncestryAI is implemented as a web tool. A screenshot is shown in Figure 1. Compared to the existing search interface [12] of the parish records, which we use as our data source, AncestryAI provides the following key improvements.

Link probabilities. Using the method described in Section 4, we rank the candidate parents of each child by their probability and visualize the most probable links as shown in Figure 1A.

Advanced search. The most probable links between the individuals allow us to query, not only by parents or by child, who are mentioned in the same birth record, but also by other relatives of the person we are looking for. For instance, we can search for people who have a cousin called *Anna*, a child called *Mats*, and a sibling called *Maria*, as shown in Figure 1E.

Shortest path, or most-probable path, between people. A user can search for a path between two people in the graph. This can be used, for example, to check whether a grandparent of the user is a relative of a historical celebrity, such as *Aleksis Kivi*, a national writer of Finland, who is found in the dataset.

Link annotation. Users can attach comments to the nodes to report an incorrectly linked individual or to confirm a link inferred by the algorithm, as shown in Figure 1C. This enables crowdsourcing training and evaluation data for the model. The comments can also help other users who see them when studying the same individual.

4. PROBABILISTIC FAMILY TREE INFERENCE

Family trees can be inferred by matching the birth records of children to the birth records of their parents. The matching of records is a challenging task, due to possible errors in the data, duplicate names, and missing records, and thus a robust computational method is needed. In this section, we develop a probabilistic model for matching the records.

4.1 Matching Probability

Consider a birth record of an individual, which we want to match with the birth record of one parent. Let M be a discrete random variable indicating the matching birth record of the parent. We want to estimate the matching probabilities over n candidate birth records of the parent; the case that the matching record is missing is denoted by $M = n+1$. The probabilities are estimated based on the similarity values between the attributes of the candidate records and the attributes of the record to be matched. For instance, if candidate record i has *Paul* as the name of the child and the record to be matched has *Paulus* as the name of the father, the similarity value of the names, γ^i_{name}, as well as the likelihood of observing such a similarity value given that the records are matched, $p\left(\gamma^i_{\text{name}} \mid M = i\right)$, will be high. The similarity values for the different attributes of candidate i are denoted by $\boldsymbol{\gamma}^i$, which is called the *comparison vector* [5, 2]. The similarity values of all n candidate records are denoted by $\boldsymbol{\gamma} = \left[\boldsymbol{\gamma}^1, \ldots, \boldsymbol{\gamma}^n\right]$. More details about the definition of the similarity values for the different attributes will follow in Section 4.3.

First note that the likelihood $p\left(\boldsymbol{\gamma}^k \mid M\right)$ of the comparison vector for candidate k depends only on whether k is the matching candidate $M = k$ or an non-matching candidate $M \neq k$. Thus the joint likelihood of the candidate vectors can be written as

$$\begin{aligned} p\left(\boldsymbol{\gamma} \mid M = i\right) &= \prod_{k=1}^{n} p\left(\boldsymbol{\gamma}^k \mid M = i\right) \\ &= \frac{p\left(\boldsymbol{\gamma}^i \mid M = i\right)}{p\left(\boldsymbol{\gamma}^i \mid M \neq i\right)} \prod_{k=1}^{n} p\left(\boldsymbol{\gamma}^k \mid M \neq k\right) \\ &= C\, \frac{p\left(\boldsymbol{\gamma}^i \mid M = i\right)}{p\left(\boldsymbol{\gamma}^i \mid M \neq i\right)}, \end{aligned}$$

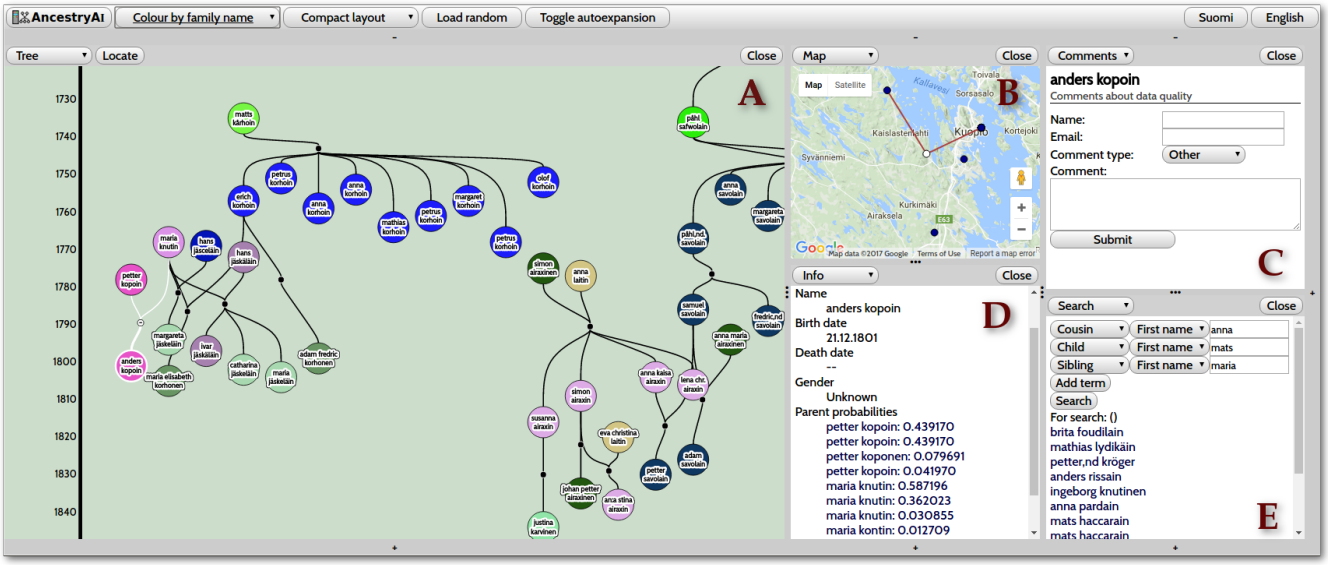

Figure 1: Screenshot of ANCESTRYAI with the following subviews: (A) Visualized family tree; (B) Birth locations of the people and the edges between the selected person (*Anders Kopoin*) and his parents and children; (C) Annotation view; (D) Information about the selected person; (E) Search view.

where the product term C is constant with respect to i. Now the matching probability is given by the Bayes' rule

$$\begin{aligned} & p\left(M=i \mid \boldsymbol{\gamma}\right) \\ &= \frac{p\left(M=i\right) p\left(\boldsymbol{\gamma} \mid M=i\right)}{\sum_{j=1}^{n+1} p\left(M=j\right) p\left(\boldsymbol{\gamma} \mid M=j\right)} \\ &= \frac{p\left(M=i\right) p\left(\boldsymbol{\gamma}^i \mid M=i\right) \,/\, p\left(\boldsymbol{\gamma}^i \mid M \neq i\right)}{\sum_{j=1}^{n+1} p\left(M=j\right) p\left(\boldsymbol{\gamma}^j \mid M=j\right) \,/\, p\left(\boldsymbol{\gamma}^j \mid M \neq j\right)}, \end{aligned} \tag{1}$$

where $p\left(\boldsymbol{\gamma}^{n+1} \mid M=n+1\right) \,/\, p\left(\boldsymbol{\gamma}^{n+1} \mid M \neq n+1\right)$ is defined to be 1. It is common to assume that the record attributes are conditionally independent [2], which allows us to write $p\left(\boldsymbol{\gamma}^j \mid M=k\right) = \prod_i p\left(\gamma_i^j \mid M=k\right)$, where the product is over the different record attributes, that is, name, birth place, and birth date in our case. The estimation of these likelihood terms along with the prior probabilities are discussed in Section 4.3.

Note that Equation (1) is closely related to the famous Fellegi–Sunter model [5], which ranks record pairs based on their likelihood ratios $p\left(\boldsymbol{\gamma}^i \mid M=i\right) \,/\, p\left(\boldsymbol{\gamma}^i \mid M \neq i\right)$, but here we additionally weight the ratios by the prior probability $p\left(M=i\right)$ and normalize them to get probabilities.

4.2 Computational Issues

Comparing every pair of records would lead to a quadratic running time, which will not be feasible when the number of records is in millions. Therefore, it is common to employ *blocking* (also referred to as *indexing*) [2], which can significantly reduce the number of record comparisons by only considering record pairs with a nonzero matching probability.

For birth record matching, the birth date is a natural blocking criteria. It is relatively safe to discard all record pairs where the candidate parent is either less than 10 years older or more than 70 years older than the child. Furthermore, we normalize names by clustering unique first names and unique last names, then assigning each full name into a first name cluster and a last name cluster. This allows us to efficiently retrieve only the people belonging to the same first and last name cluster. Clustering is done simply by going through the list of unique names once, assigning name A to the cluster whose representative name has the highest Jaro–Winkler similarity with A given that the similarity is above a threshold of 0.9. Otherwise, A is assigned to a new cluster whose representative name is fixed to A.

The records with suitable age difference and matching name clusters, which are obtained after the blocking step, are called *candidate matches*.

4.3 Model Estimation

To estimate the terms of Equation (1), we use ground-truth data with known matches. The prior probability of a missing match $p\left(M=n+1\right)$ can be estimated based on the fraction of training records for which the golden match is not found among the candidate matches.

To compute the comparison vector $\boldsymbol{\gamma}$, we need to define a similarity measure for each record attribute. For name comparisons, we use the Jaro–Winkler string similarity, since Jaro–Winkler is a popular choice for de-duplicating name records [13]. For birth place comparisons, we can use the distance between their coordinates, which are estimated using the method described in Malmi et al. [8]. Finally, for birth year comparisons, we use the difference between the birth years.

The likelihood term $p\left(\gamma_i^j \mid M=j\right)$ can be computed by the distribution of the similarity measure values of attribute i among the golden matches. For the likelihoods of the non-matching pairs $p\left(\gamma_i^j \mid M \neq j\right)$, golden matches are not needed. These distributions can be computed based on the similarity measures among all candidate matches for a sample of birth records.

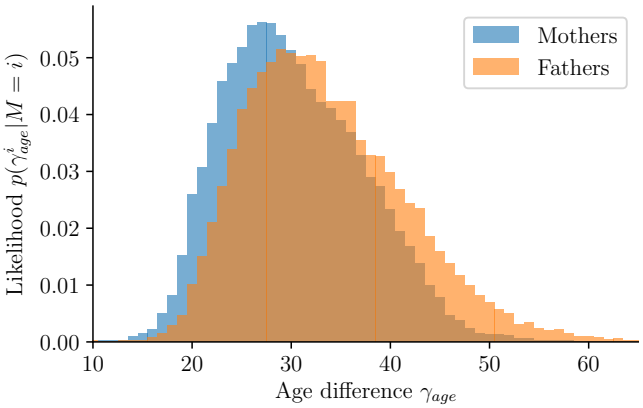

Figure 2: Likelihood distributions for the age difference between a newborn child and his or her parents. The average age of mothers is 30.3 years and of fathers 33.5 years.

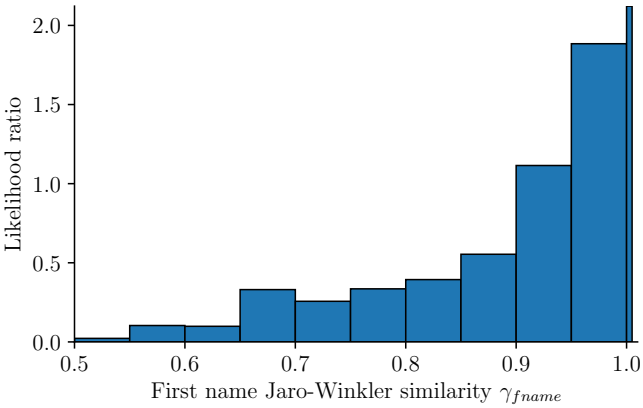

Figure 3: Likelihood ratios $p\left(\gamma^i_{fname} \mid M = i\right) \;/\; p\left(\gamma^i_{fname} \mid M \neq i\right)$ for different first name similarities. The rightmost bar shows the ratio when the similarity is exactly 1.

The likelihood distributions for the age difference between matching records are shown in Figure 2. The likelihood ratios given the first name similarity $p\left(\gamma^i_{fname} \mid M = i\right) \;/\; p\left(\gamma^i_{fname} \mid M \neq i\right)$ are shown in Figure 3. If the first name similarity is 1 (i.e., the names are identical), it doubles the probability of the candidate record to be the true match, whereas if the similarity is between 0.85 and 0.9 it halves the probability.

5. FAMILY TREE LAYOUT ALGORITHM

Genealogical data is often visualized by drawing a tree of ancestors or a tree of descendants relative to a root individual. In our case, the aim is to provide a visualization, which is not tied to an individual root node and which can be easily explored. Additionally, it is preferable to minimize the number of edge crossing, although this typically leads to an **NP**-hard problem [6], since in the graph-theoretic sense, family trees are directed acyclic graphs rather than trees.

Next we describe the layout algorithm used in AncestryAI. This algorithm is based on relatively simple heuristics, which allows it to run on the client, making the system more scalable. The user of the system is first shown a single individual. The user can expand the graph by clicking on unopened nodes. The layout of the graph is dynamically adjusted using smoothly animated transitions.

The y–coordinate of each individual is fixed based on the birth year of the person. If the birth year is unknown, it is estimated based on the birth years of the neighbors or set to a default value if the neighbors are unknown.

The x–coordinate is based on arranging the individuals into a one-dimensional array, which is updated every time an unopened node is clicked which loads the parents and the children not yet shown. Ignoring some details, the idea is to place a newly-loaded node A to the right of the rightmost descendant of A's parents. If the parents are not yet loaded, it will be placed to the left of the leftmost children of A.

At times, when expanding node B, we end up loading node C who is a parent, a child, or a spouse of an already loaded node $D \neq B$. This happens due to intermarriage, which causes multiple paths between nodes. As a result, a child may appear before its parents in the array, which is fixed by repositioning the child and its children recursively according to the aforementioned rules.

The array indices of the loaded individuals could be used as the x–coordinates directly but we additionally shift the individuals to the left if there is empty space, allowing nodes to be vertically aligned to fill the space more efficiently.

For family trees that only contain a limited number of contemporaries, this approach produces easy-to-read visualizations. When the trees grow wider, we observe longer and harder-to-follow edges, which cannot, however, be avoided altogether when minimizing edge crossings [9]. To make the tree more readable in these cases, we added the option to collapse the subtree of any individual after receiving the feedback from genealogists who tested an earlier version of AncestryAI.

Finally, note that it is straightforward to adopt any alternative approach for determining the coordinates of the individuals as long as it is feasible to compute them on the client-side.

6. RELATED WORK

Record linkage for genealogical data has been previously studied, e.g., by Efremova et al. [4] and by Christen et al. [3]. Alternative family tree visualization approaches have been proposed by McGuffin and Balakrishnan [9] and by Bezerianos et al. [1].

7. FUTURE WORK

AncestryAI has been developed in collaboration with genealogists and we are planning to officially launch it to a wider audience during Spring 2017. Once we have collected a sufficient amount of user annotations, we plan to perform a systematic evaluation of the matching performance.

One limitation of the inference algorithm described in Section 4 is that it considers the matching problems independently. This results in a single person having children with multiple spouses with similar names, although, in reality, the children probably have the same two parents. We are currently looking into collective entity resolution methods to avoid this problem. The matching algorithm can be further improved by also matching marriage, burial, and migra-

tion records, whereas currently we only consider the birth records.

Acknowledgments

We would like to thank the Genealogical Society of Finland and Pekka Valta for providing the genealogy data. This work has been supported by the Academy of Finland project "Nestor" (286211) and the EC H2020 RIA project "SoBigData" (654024).

8. REFERENCES

[1] A. Bezerianos, P. Dragicevic, J.-D. Fekete, J. Bae, and B. Watson. Geneaquilts: A system for exploring large genealogies. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1073–1081, 2010.

[2] P. Christen. *Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection*. Springer, 2012.

[3] P. Christen, D. Vatsalan, and Z. Fu. Advanced record linkage methods and privacy aspects for population reconstruction—a survey and case studies. In *Population Reconstruction*, pages 87–110. Springer, 2015.

[4] J. Efremova, B. Ranjbar-Sahraei, H. Rahmani, F. A. Oliehoek, T. Calders, K. Tuyls, and G. Weiss. Multi-source entity resolution for genealogical data. In *Population Reconstruction*, pages 129–154. Springer, 2015.

[5] I. P. Fellegi and A. B. Sunter. A theory for record linkage. *Journal of the American Statistical Association*, 64(328):1183–1210, 1969.

[6] M. R. Garey and D. S. Johnson. Crossing number is NP-complete. *SIAM Journal on Algebraic Discrete Methods*, 4(3):312–316, 1983.

[7] D. Lazer, A. S. Pentland, L. Adamic, S. Aral, A. L. Barabasi, D. Brewer, N. Christakis, N. Contractor, J. Fowler, M. Gutmann, et al. Life in the network: the coming age of computational social science. *Science*, 323(5915):721–723, 2009.

[8] E. Malmi, A. Solin, and A. Gionis. The blind leading the blind: Network-based location estimation under uncertainty. In *Proc. ECML PKDD*, pages 406–421. Springer, 2015.

[9] M. J. McGuffin and R. Balakrishnan. Interactive visualization of genealogical graphs. In *Proc. INFOVIS*, pages 16–23. IEEE, 2005.

[10] J. E. Pettay, M. Lahdenperä, A. Rotkirch, and V. Lummaa. Costly reproductive competition between co-resident females in humans. *Behavioral Ecology*, pages 1–8, 2016.

[11] E. Salmela, T. Lappalainen, J. Liu, P. Sistonen, P. M. Andersen, S. Schreiber, M.-L. Savontaus, K. Czene, P. Lahermo, P. Hall, and J. Kere. Swedish population substructure revealed by genome-wide single nucleotide polymorphism data. *PLoS One*, 6(2):e16747, 2011.

[12] The Genealogical Society of Finland. HisKi project (Web interface). `http://hiski.genealogia.fi/hiski?en`, Accessed: 2017-01-07.

[13] W. E. Winkler. String comparator metrics and enhanced decision rules in the Fellegi–Sunter model of record linkage. In *Proceedings of the Section on Survey Research Methods*, pages 354–359. American Statistical Assn., 1990.

[14] E. Zagheni, V. R. K. Garimella, I. Weber, et al. Inferring international and internal migration patterns from twitter data. In *Proc. WWW*, pages 439–444. ACM, 2014.

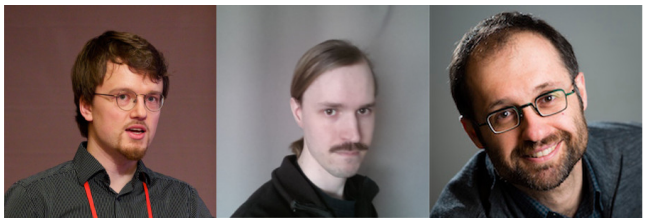

Figure 4: The authors from left to right: Eric Malmi, Marko Rasa, and Aristides Gionis.

Eric Malmi is a doctoral student in the Department of Computer Science at Aalto University. Previously, he has done internships at Google Research, Qatar Computing Research Institute, Idiap Research Institute, and CERN. In his doctoral thesis, he studies data mining and machine learning methods for the analysis of historical and social media data.

Marko Rasa is a software developer at Verto Analytics. He holds a M.Sc. (Tech.) from Aalto University and has worked there as a research assistant in the Department of Computer Science.

Aristides Gionis is an associate professor in the Department of Computer Science in Aalto University. Previously he has been a senior research scientist in Yahoo! Research. His research interests include several areas of data science, such as graph mining, social-media analysis, web mining, data clustering, and urban computing.