An Approach to Population Linkage using Graph Databases

Alan Dearle[1], Graham Kirby[1] and Özgür Akgün[1]

[1]*School of Computer Science, University of St Andrews, St Andrews, KY16 9SX, Scotland*

Abstract
We report on a database project which is in the process of linking 29 million vital event records encompassing the entire population of Scotland from 1856 until 1973. Since these records contain no common identifiers, the challenge is to form a pedigree by performing probabilistic linkage over the records. We describe the linkage methodology used to create links between records, for example identifying the birth and marriage records of a single person, and discuss the database technologies employed in the project. A graph database (Neo4j) is used to store both the original vital event records and the links made between them. A metric index is used to find potential links efficiently. Finally, we demonstrate how linkage can be improved by augmenting links based on record distance thresholds with local graph analysis.

Keywords
data linkage, graph databases, similarity search, metric indexing, metric search

1. Introduction

The SHiPP Scotland project [1] is in the process of linking the contents of the civil registers of births, marriages and deaths for Scotland covering 1856-1973. This project aims to undertake a family reconstitution exercise which will encompass the entire population of Scotland over these 12 decades: some 14 million births, 11 million deaths and 4 million marriages.

The records have now been transcribed, and must be linked in order to reconstruct the population structure. This involves deciding which particular subsets of records should be linked, and in what way, and selecting appropriate linkage algorithms. Although this is not a massive dataset by modern standards it does present a database challenge, since complex data relationships must be established by linkage algorithms. The output of this project, intended primarily as a resource for further research in the social sciences, will be linked pedigrees containing the individual people represented in the civil records, along with the relationships between them. This paper will describe an approach to establishing these relationships.

SEBD 2023: 31st Symposium on Advanced Database Systems, July 02–05, 2023, Galzignano Terme, Padua, Italy
alan.dearle@st-andrews.ac.uk (A. Dearle); graham.kirby@st-andrews.ac.uk (G. Kirby); ozgur.akgun@st-andrews.ac.uk (Ö. Akgün)
0000-0002-1157-2421 (A. Dearle); 0000-0002-4422-0190 (G. Kirby); 0000-0003-4734-8103 (Ö. Akgün)

2. Population Data

In the period of study, the Scottish birth records collected by the General Register Office for Scotland include the following fields (reproduced from [2]):

- register entry number, year, registration district number and suffix, child's forename(s) and surname, child's sex, date and place of birth, mother's forename(s), surname and maiden surname, father's forename(s) and surname, parents' date and place of marriage, father's occupation.

Death records include the following fields:

- register entry number, year, registration district number and suffix, deceased's forename(s) and surname, deceased's sex, date, place and cause of death, deceased's date of birth (or their age at death), deceased's occupation, deceased's marital status, deceased's spouse's name and occupation, deceased's mother's forename(s), surname and maiden surname, deceased's father's forename(s) and surname, whether deceased's parents were deceased.

Marriage records include the following fields:

- register entry number, year, registration district number and suffix, groom's forename(s) and surname, bride's forename(s) and surname, date and place of marriage, religious denomination, bride and groom's dates of birth (or their ages at marriage), bride and groom's addresses, bride and groom's occupations, bride and groom's previous marital status, bride and groom's mothers' and fathers' forenames, surnames and maiden surnames, bride and groom's fathers' occupations, whether bride and groom's parents were deceased.

3. Linkage Methodology

Probabilistic linkage is the process by which entries in database records may be determined to be related to the same underlying entity [3, 4], without any common identifier. A probabilistic linkage process must account for errors, inconsistencies and omissions in the data, and also ambiguity where there are alternative possible links.

We distinguish between *entity linkage*, in which a single individual is identified as appearing in multiple records, and *relationship linkage*, in which relationships between different individuals appearing in multiple records are established. Our approach involves both entity linkage and relationship linkage. Examples include:

- **entity linkage**: linking a woman's birth record to her marriage records, or linking a man's birth record to the death records of his daughters
- **relationship linkage**: linking the birth records of full siblings

The amount of relevant information available on which to make linking decisions varies with the type of link. In the first entity linkage example above, the fields containing the woman's mother's names and her father's names might be compared with the mother's and father's names on a marriage record. The relationship linkage example might be performed by comparing the values of the fields containing the parents' names and the places and dates of their marriages.

3.1. Comparing Records

Comparator functions are used to determine the distance (i.e. lack of similarity) between two records, by comparing corresponding pairs of fields drawn from the vital event records. Each comparator function operates over selected fields of the records. A *base comparator* compares a single field of one record with a single field of another record. A *composite comparator* compares two records by combining a set of base comparators, perhaps each with a separate weighting. For example, a simple composite comparator for comparing birth and death records—to link the birth records of individuals to their own death records—may be defined as follows:

$$\begin{aligned}&\mathit{composite\text{-}comparator}(birth, death) = \\ &\quad \mathit{base\text{-}comparator}(birth.forename, death.forename) + \\ &\quad \mathit{base\text{-}comparator}(birth.surname, death.surname)\end{aligned}$$

We employ *true metrics* as comparators, in order to access the power of metric indexing [5]. A metric is a comparator that satisfies the postulates of non-negativity, identity, symmetry, and triangle inequality. This excludes some functions that are often referred to as distance metrics (such as Jaro-Winkler [6]), but do not satisfy the triangle inequality, and are therefore not true metrics. We discuss the use of metric indexing techniques in Section 5.

Initial linkage decisions are made by comparing the distance between each candidate pair of records, calculated by the appropriate composite comparator, with a predetermined distance threshold. Thresholds for each type of linkage are calibrated using known ground-truth examples. Initial linkage decisions are refined later in the process, as discussed in Section 6.

3.2. Linkage Opportunities

Each record contains information on a different number of individuals, depending on the record type:

- **three** on a birth record (child, mother, father)
- **six** on a marriage record (bride, groom, two sets of parents)
- **four** on a death record (deceased, spouse, deceased's parents)

Figure 1 lists the possible combinations for **entity linkage** between individuals on two records.

- The combinations shown in red are logically not possible, since at most one birth and death event can be recorded for an individual. For example: the child on one birth record cannot be linked to the child on another birth record.
- The combinations shown in pink are also logically excluded, due to the sexes of the individuals. For example: the groom on a marriage record cannot be linked to the mother on a birth record.
- The combinations shown in green are logically possible. Those shown in dark green involve a significant amount of common information between the records, allowing links to be made with greater confidence than those shown in light green. We denote the former *strong linkage opportunities*.

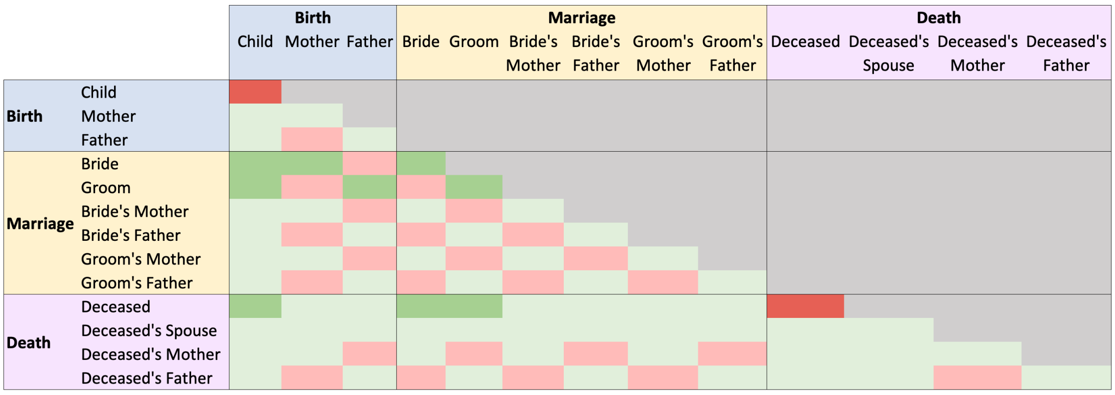

Figure 1: Entity linkage opportunities

Overall, we identify 64 different types of logically possible entity linkage.

There are many logically possible types of **relationship linkage**. It is not necessary to perform linkage to establish the most straightforward types, such as parent-child and spouse-spouse links, since they are captured within a single record. Conversely, for relatively obscure linkages such as linking a person's birth record to their great-aunt's death record, there is unlikely to be sufficient common information to make the linkage feasible. The most practical type of relationship linkage is sibling linkage: while siblings are not recorded on any single record, there is a significant amount of information in common between the birth, death or marriage records of siblings.

Figure 2 lists the opportunities for relationship linkage between siblings on two records. While all combinations are logically possible, the feasibility of establishing links varies with the amount of common information. Those shown in dark green involve include the names of both sets of parents for a potential sibling pair, in addition to temporal and geographical information. The combinations shown in light green are less likely to be feasible to establish, since less common information is available.

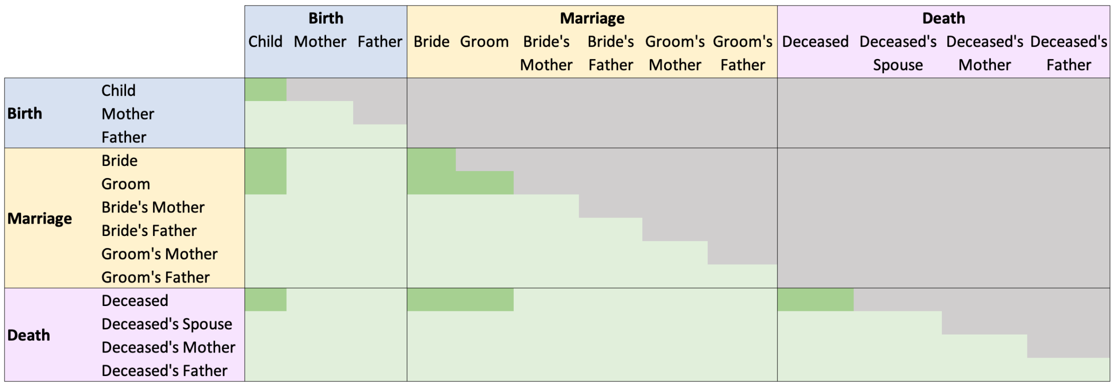

Figure 2: Relationship (sibling) linkage opportunities

4. The Linkage Process

In our prototype, we define a number of aspects common to the various types of linkage, in a generic *linker* function. This takes as input either one (when linkage is over a single type of event record) or two sets of records (when linkage is over different types of event record). For example, one set of records is supplied for linkage of birth records of siblings or linkage of the births of children to the births of their mothers. Two sets are supplied for linkage of the births of children to their death records.

The output of the linker is a set of record pairs for which the distance between the records is below some defined threshold. Each pair is annotated with the type of linkage. A specialised linker for a particular type of linkage is obtained by configuring the generic linker with a *recipe* defining the aspects specific to that linkage:

- the types of the records
- the source(s) of the records (for example, particular files, or a database)
- the fields to be compared
- the distance threshold
- any link viability constraints (for example, temporal rules such as death occurring after birth, or minimum age at marriage)

For example, *birth-bride-entity* linkage involves linking the birth records of female children to their marriage records. The recipe includes the following:

- **types**: birth records, marriage records
- **sources**: database queries to retrieve the records
- **fields**:
  - birth records: *forename*, *surname*, *mother_forename*, *mother_maiden_surname*, *father_forename*, *father_surname*
  - marriage records: *bride_forename*, *bride_surname*, *bride_mother_forename*, *bride_mother_maiden_surname*, *bride_father_forename*, *bride_father_surname*
- **threshold**: 0.78 (as determined by previous experimentation)
- **constraints**: child sex is female, marriage date after birth date, difference between dates greater than 14 years and less than 120 years

The first stage of the linkage process involves running a linker for each of the linkages highlighted in dark or light green in Figures 1 and 2. These initial linkage results are then refined as discussed in Section 6, and finally combined (not discussed in this paper).

5. Database Requirements

This project has some unique database requirements:

1. the storage of the original 29 million source records,
2. the ability to efficiently search the records by distance (for various definitions of distance) in order to find records for linking,
3. the storage of the links formed between these records, along with meta data including the linker used, distances between records and/or probabilities of each link being correct, and
4. (as described below) the ability to navigate the database to analyse and improve linkage quality.

For requirements 1 and 3, any traditional relational database could be employed. This would not be suitable for requirement 4 however, due to the inability of relational systems to perform transitive queries efficiently, while requirement 2 cannot be met by any conventional database.

To address requirements 1, 3 and 4 we use the Neo4j graph database [7]; to address requirement 2 we employ a *metric index*, BitPart [8]. This addresses the issue that comparing each record with every other record would be prohibitively expensive. *Blocking* [4] is sometimes employed to avoid polynomial complexity, but it has the disadvantage of unavoidable false negatives [5].

The BitPart metric index creates a set of *inclusion zones* encoding the inclusion of data points in a set of database partitions in a binary fashion with respect to a set of *reference points*. A relatively small number of reference points (in the order of 20-40) is enough to characterise the search space, and each metric query only requires distances to be calculated to the reference points. For large datasets such as the Scottish vital event records, this represents a large performance increase. Furthermore, the index is highly compressed (as a set of bits) and obviates the need to directly interact with the stored database records when making queries.

When linkage is performed between two datasets, the larger set is loaded into the metric index. The composite comparator and the threshold are supplied. Each record from the smaller set is used to query the BitPart index, and any records with a distance from the query term below the threshold are retrieved and recorded in the database, as described below.

We use Neo4j to store both the original source records and the links formed between them. Such a database is well suited for this purpose, as it supports attributed relationships between nodes of arbitrary type. The original vital event records (birth, death and marriage records) are stored as Neo4j nodes containing the field data from the original records. Thus the database initially contains a number of unconnected data records containing the original data. The linkage process involves linking these records by creating relationships between them.

Once a linker has found links between records using the BitPart index, it encodes the links between these original records as relationships between the stored nodes in the graph database. Each of the relationships has attributes which encode the name of the linker used to form the link (and thus all the provenance in the source code), the distance between the nodes (from the metric search), and the type of relationship (for example mother, father, sibling, entity (same individual on two records)). In some cases additional information is stored in the relationship attributes to disambiguate the links established (for example, in the case of entity links, which individuals on the records are involved). Figure 3 shows a link formed between a birth record and a corresponding death record. The relationship shows the metric distance between the records, and the record identifiers (571766 and 571764).

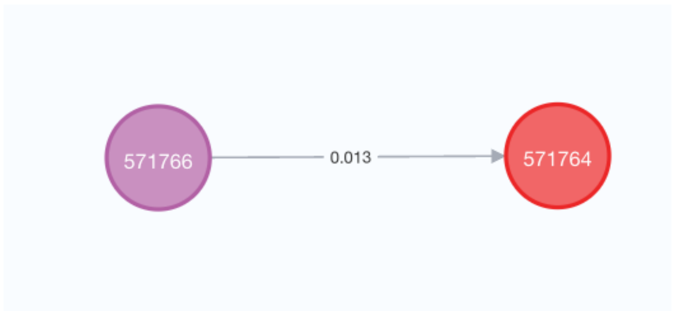

Figure 3: A birth-death link

In addition to the ease with which linkage information may be encoded, Neo4j facilitates querying such graph structures, using the rich Cypher query language. For example, the nodes shown in Figure 3 could be retrieved via a number of queries specifying varying degrees of detail, as shown in Figure 4.

```
MATCH (d:Death) WHERE ID(d) = 571764 RETURN d;
MATCH (b)-[r]-(d) WHERE r.linker = "Birth-Death-Entity" RETURN b,r,d;
MATCH (b:Birth)-[r]-(d) WHERE r.distance = 0.013 RETURN b,d;
MATCH (b:Birth)-[r]-(d:Death) RETURN b,r,d;
MATCH (b:Birth)-[r linker:"Birth-Death-Entity"]-(d:Death) RETURN b,r,d;
```

Figure 4: Example Cypher queries

The first example matches a node based on its identity; the second matches based on the *linker* attribute of a relationship; the third uses the *distance* attribute of a relationship. Note that the types of the nodes may be specified, or not, and that the queries may return multiple nodes, relationships or a combination of both. The final example returns all nodes and relationships between birth and death records that were created by the *Birth-Death-Entity* linker.

Once all linkers have completed, the original graph containing only the source records has been transformed into a labelled graph containing relationships between the nodes. Figure 5 shows a simplified example of the output, omitting many of the links. The purple and blue nodes in the diagram represent birth and marriage records respectively. The purple nodes denote the birth records of siblings, linked with *child-child* relationships established by a *Birth-Birth-Sibling* linker. Each birth record is linked to a marriage record by (redundant) *mother-bride*, *father-groom* and *child-couple* relationships. Such redundant links may seem wasteful, but they can be formed by different linkers, comparing different fields, thus adding confidence (or otherwise) to the links that have been formed. In the diagram, the nodes are labelled with the identity of the father of the children and the groom in the marriage record. We can see that, in this case, since the identities match, the linkers have made consistent assertions about the relationships. In this example the ground truth is extracted from a known dataset.

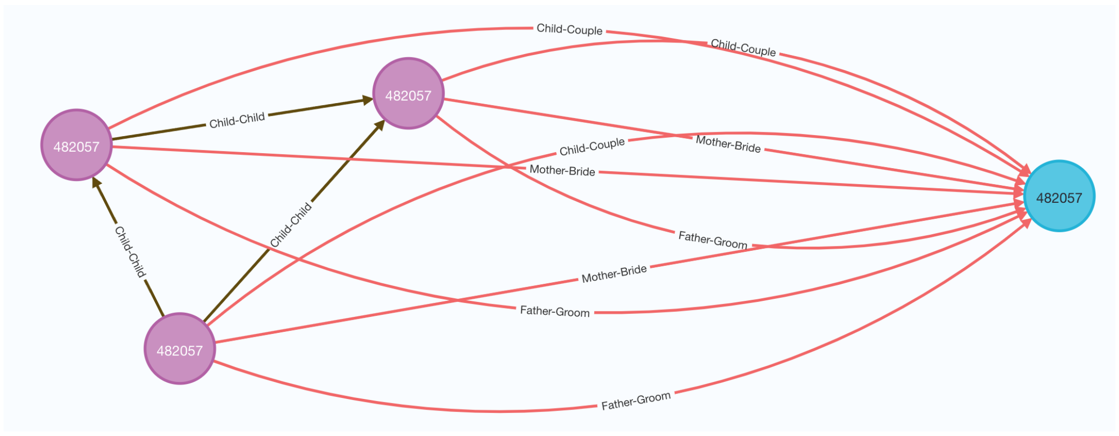

Figure 5: Example links

6. Linkage Refinement

Linkage is followed by a refinement process which aims to improve the quality of the linkage. The graphs that are created in the linkage phase may contain various classes of error:

1. **Errors of omission**: where a link should exist in the linkage graph, but the linkers have not established the link.
2. **Errors of inclusion**: where a link has been created in error. For example, a cluster of siblings may form one graph but in fact represents multiple families.
3. **Uniqueness constraint errors**: where a single relationship should exist in the graph but more than one has been established. For example, if entity links are established between the child on one birth record and multiple death records, at least one link must be in error.

6.1. Detecting Errors

We use Cypher queries to detect such errors. In some cases we can then decide how to rectify the error based on information in the retrieved nodes and their neighbours in the relationship graph.

Some errors can be detected by considering transitive relationships, and searching for cases where the logical consequences of transitivity are not reflected in the graph. For example, the relationship *is-a-sibling-of* is transitive: if person A is a sibling of person B, and B is a sibling of C, then A must be a sibling of C. Similarly, if a person recorded on record X is the same person as one recorded on record Y, and the person recorded on Y is the same person as one recorded on record Z, then the person recorded on X must be the same person as recorded on Z.

The simplest sub-graph to which a transitivity check can be be applied is a triangle of three nodes:

- If the sub-graph is not connected, i.e. at least one node is not connected to another, the triangle can be ignored.
- If the sub-graph is fully connected, the transitivity condition is satisfied, and no error is apparent.
- If the sub-graph contains exactly two relationships (edges), we consider this an *open triangle*, which indicates an inconsistency in transitivity, arising from either an error of omission or an error of inclusion.

Open triangles can be identified via a simple Cypher query, as illustrated in Figure 6.

```
MATCH (x:Birth)-[:SIBLING]-(y:Birth)-[:SIBLING]-(z:Birth)
WHERE NOT (x)-[:SIBLING]-(z) RETURN x,y,z
```

Figure 6: A query to search for open triangles in sibling linkage over birth records

Figure 7 shows an example of an open triangle.

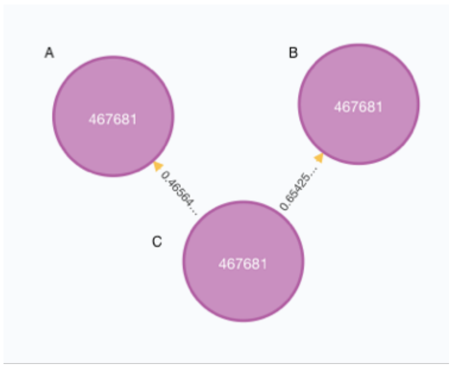

Figure 7: An open triangle

Open triangles may contain nodes of the same type (e.g. three birth records representing siblings), or nodes of different types (e.g. a birth, death, and marriage record all representing the same person). More complex patterns of non-transitivity may also be identified, for example chains of nodes that are not fully connected. Searching for open triangles is sufficient to identify such patterns, although consideration of the surrounding graph may be beneficial in deciding how to rectify errors.

For some types of linkage we can define constraints such that a record should not be linked to more than one other record. Examples of potential errors include:

1. Multiple births corresponding to one death
2. Multiple deaths corresponding to one birth
3. Multiple parents' marriages corresponding to one child birth
4. Multiple bride births/deaths corresponding to a single marriage

Such errors may be identified via Cypher queries such as that illustrated in Figure 8.

```
MATCH (b:Birth)-[r linker:"Birth-Death-Entity"]->(d:Death) WITH d, count(r) as countlinks
WHERE countlinks > 1
RETURN d;
```

Figure 8: A query to search for multiple deaths linked to a single birth

6.2. Correcting Errors

We cannot determine with certainty whether an open triangle indicates an error of omission or inclusion. In the example of Figure 7, it is possible that there should be a link between A and B (omission), or that there should not be links between A and C, and B and C (inclusion). Given that two out of three links have been established, it may be reasonable to conclude that the error of omission is more likely. The distances AC and BC may also be considered: if they are relatively high and close to the threshold, an error of inclusion may be more likely.

We can also look for potential graph isomorphisms when we consider entity and relationship linkage graphs together. For example, Figure 9 shows an open triangle for sibling linkage using birth records. Since an entity link has also been established between each birth record and a corresponding death record, and those death records form a complete triangle for sibling linkage, we can have greater confidence that the missing birth sibling link should be present.

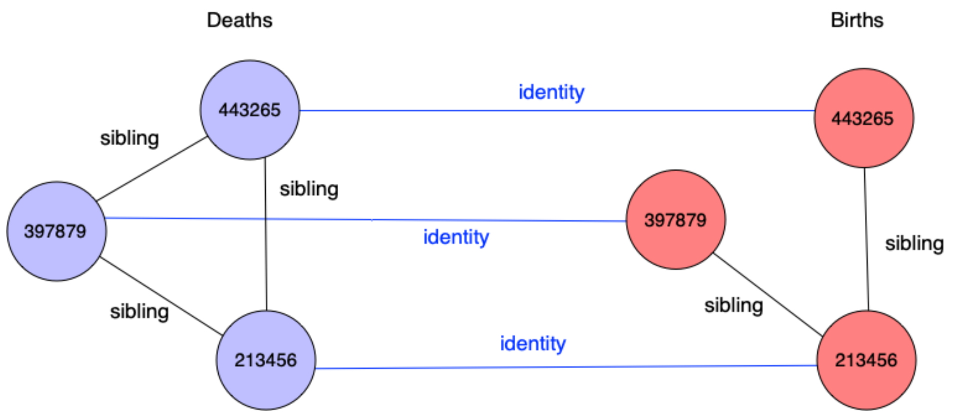

Figure 9: Incomplete sub-graph isomorphism indicating error of omission

In the example above, the presence of entity links is used to support the existence of missing relationship links. Conversely, if both relationship triangles were complete but one of the entity links was missing, its existence could be inferred.

7. Conclusions

We have reported on a project to link 29 million vital event records. We have sketched how the linkage code operates and its use of two different database technologies. A state of the art metric index is used to perform distance calculations over the records and to determine which records should be linked. A graph database is used to store both the original record and the links created between them, and to facilitate analysis of the graph structures to refine linkage decision. The project code is published [9]. Work continues on aspects including:

1. **Combining linkages**: we are investigating approaches to synthesising a coherent population structure from the results of the large number of different linkages (74 linkage types were identified in Section 3). We aim to exploit the high degree of redundancy in the structure of the data, in that genealogical relationships are encoded in multiple ways (for example, sibling relationships may be deduced independently from birth, death and marriage records). Although some of the types of linkage are difficult to perform in isolation due to a limited amount of common information between the records, we take encouragement from the success of ensemble methods in machine learning; multiple weak linkages may still improve overall results.
2. **Calibrating parameters**: our approach requires various parameters to be configured, in particular the distance thresholds for each type of linkage. In order to establish appropriate parameter values we need ground truth data against which linkage quality can be optimised. The scarcity of ground truth is a general problem in automated approaches to population reconstruction, due to the relatively low number of available population datasets and the high cost of expert annotation. Furthermore, even when available, such ground truth may be incomplete, biased, and contain significant numbers of errors [10]. We are fortunate to have access to a relatively large ground truth dataset from the CEDAR group in Umeå, Sweden [11]. This data is encoded in a similar form to the Scottish data and thus permits us to experiment with some confidence that results can be extrapolated to the Scottish data. Another promising avenue to obtaining ground truth is statistically faithful simulation of large-scale populations [10, 12].
3. **Evaluating approaches**: similar to the calibration issues discussed above, it is also desirable to be able to experiment with various linkage approaches. Again, access to realistic ground truth data at scale is essential to evaluation.
4. **Delivering results**: the simplest approach to delivery of a complete population reconstruction would be to collate all of the final links into a unified genealogical structure, apply any necessary anonymisation, and make this data available to approved researchers. However, the utility of this resulting data would be dependent on having made optimal decisions on linking thresholds and graph refinements throughout the process. Furthermore, the trade-off between Type 1 and Type 2 errors in the population structure might, ideally, be set differently depending on the research question. We are considering approaches to allowing some degree of control over this trade-off by the end-user researcher.

Acknowledgments

This work was supported by ESRC grants ES/K00574X/2 "Digitising Scotland", ES/L007487/1 "Administrative Data Research Centre – Scotland", ES/S007407/1 "Administrative Data Research Centres 2018" and ES/W010321/1 "2022-2026 ADR UK Programme".

We thank our collaborators on the SHiPP project: Chris Dibben, Peter Christen, Lee Williamson and Eilidh Garrett. We are also indebted to our colleagues in Umeå University who created the ground truth used in this project, especially Maria Larsson and Pär Vikström.

References

[1] SHiPP, Scottish Historic Population Platform (SHiPP), 2023. URL: https://www.scadr.ac.uk/our-research/shipp.

[2] Ö. Akgün, A. Dearle, G. Kirby, E. Garrett, T. Dalton, P. Christen, C. Dibben, L. Williamson, Linking Scottish Vital Event Records using Family Groups, Historical Methods: A Journal of Quantitative and Interdisciplinary History (2019) 1–17. URL: https://doi.org/10.1080/01615440.2019.1571466.

[3] I. P. Fellegi, A. B. Sunter, A Theory for Record Linkage, Journal of the American Statistical Association 64 (1969) 1183–1210. URL: https://doi.org/10.1080/01621459.1969.10501049.

[4] P. Christen, Data Matching: Concepts and Techniques for Record Linkage, Entity Resolution, and Duplicate Detection, Springer Publishing Company, Incorporated, 2012. URL: https://doi.org/10.1007/978-3-642-31164-2.

[5] Ö. Akgün, A. Dearle, G. Kirby, P. Christen, Using Metric Space Indexing for Complete and Efficient Record Linkage, in: D. Phung, V. S. Tseng, G. I. Webb, B. Ho, M. Ganji, L. Rashidi (Eds.), Advances in Knowledge Discovery and Data Mining, Springer International Publishing, 2018, pp. 89–101. URL: https://doi.org/10.1007/978-3-319-93040-4_8.

[6] W. Winkler, String Comparator Metrics and Enhanced Decision Rules in the Fellegi-Sunter Model of Record Linkage, Proceedings of the Section on Survey Research Methods (1990). URL: https://eric.ed.gov/?id=ED325505.

[7] Neo4j, Neo4j Graph Data Platform, 2023. URL: https://neo4j.com.

[8] A. Dearle, R. Connor, Bitpart: Exact Metric Search in High(er) Dimensions, Information Systems 95 (2021) 101493. URL: https://doi.org/10.1016/j.is.2020.101493.

[9] A. Dearle, G. Kirby, T. Dalton, Population Linkage, 2023. URL: https://github.com/stacs-srg/population-linkage.

[10] T. Dalton, Evaluating Data Linkage Algorithms with Perfect Synthetic Ground Truth, Ph.D. thesis, University of St Andrews, 2022. URL: https://doi.org/10.17630/sta/247.

[11] M. J. Wisselgren, S. Edvinsson, M. Berggren, M. Larsson, Testing Methods of Record Linkage on Swedish Censuses, Historical Methods: A Journal of Quantitative and Interdisciplinary History 47 (2014) 138–151. URL: https://doi.org/10.1080/01615440.2014.913967.

[12] T. Dalton, G. Kirby, A. Dearle, Ö. Akgün, ValiPop: a Micro-simulation Model for Generating Synthetic Genealogical Populations, 2023. URL: https://github.com/stacs-srg/population-model.