Anthropologists and other researchers often study kin and other social relationships. There have been numerous applications that have tried to meet the needs of researchers in this area. However, there remain numerous gaps in either the functionality, usability or affordability of existing software. Some of these needs may be as simple as quickly creating diagrams. Some are more complex, such as statistical analysis of large sets of data. Interoperability is also a crucial factor, so that data collected in one application can be transferred to another. Linking resource files or archived material to genealogies also gives the opportunity to tie together all the data that might be available – finding individuals in a genealogy or matching a given affiliation pattern, such as matrimonial rings or other kinship patterns. Finding archived material associated with an individual or with a given affiliation pattern can also greatly assist the research process.

With these issues in mind, KinOath Kinship Archiver is a recently written application for the collection of kinship data and the subsequent exploration of that kinship data. The primary goal of this appli-

cation is to be flexible and culturally nonspecific, so that the complex facets in kinship and other social structures can be adequately represented. Beyond this, the application is designed to connect kinship data with archived data, such as audio data, video data or written resources. Kin type strings – notations for describing kinship relations – are used throughout the application for constructing and searching data sets. The representation and recording of kin terms is also integrated into the application, allowing comparative diagrams of kin terms across cultures or parallel kin term systems within the same culture. Graphical representation of the data is an important part of the application and the diagrams produced are intended to be flexible and of publishable quality.

The structure of this paper is as follows. Following consideration of some background issues behind KinOath, we discuss the internal data structure used and methods of data entry, and the various uses of kin type strings and the diagram types that are available. We then discuss methods of searching through the kinship data, the importing and exporting of kinship data, and the plugins framework that can allow extensions to be made for the application. Methods of doing statistical analysis on the data from KinOath are considered, as are the future possibilities of archiving with kinship data and the visualisation of the kinship data and the graphics used in KinOath. The conclusion includes discussion of future directions of the application.

Background

The following section describes the initial development stages of KinOath, its intended users, the relevance of kinship data to archiving, and existing applications.

Initial development

KinOath Kinship Archiver was developed by the author, Peter Withers, at the Language Archive (TLA), funded by the Max Planck Society and the Berlin-Brandenburg Academy of Sciences and Humanities. The needs assessment phase of development was done by Withers with the
assistance of researchers at the Max Planck Institute for Social Anthropology and the Max Planck Institute for Psycholinguistics.

**Intended users**

The intended users of KinOath are any researchers that collect data in a context of social relations, which could be, for instance, genetic, religious, monetary or political. Potential users could include anthropologists, linguists, ethnomusicologists, or any researcher that collects field data on interconnected individuals. The application could also be of use when the social context (e.g., social relations, genealogical relations, religious relations, ownership) is pertinent to the data collected or investigated. Geneticists may find it useful in recording the genealogical structures of a community when they are collecting genetic samples. Field data that has been collected with the inclusion of kinship relations can then be aggregated to form a stronger corpus by combining the data from each of these different fields in an aggregated kinship-based search.

**Relevance of kinship data**

Kinship data is often not systematically included in the metadata of archives and hence it could be asked: why is the kinship data relevant to archiving? When interpreting data, the context of that data can assist its interpretation. If the data collected comes from a context with social and genetic relations, then that context can be very relevant to the interpretation of the data. Even if such relevance is not immediately obvious, there may be no other opportunity to collect that information. Some of the contextual information might seem obvious at the time of collection, but to researchers interpreting that data at a later date it might not be so clear. Yet, without that social context, the data can be of reduced value or just much harder to find. By having the kinship data associated with the field data as it is archived, or preferably as it is collected, it will be possible to do archival queries based not only on the immediate metadata but also to perform kinship-based archive searches that include relatives or community members associated to that initial data point of interest.
There are many existing kinship applications available. However, many of these will be out of financial reach of students or under-resourced researchers, while others will not be able to describe the cultural facets of interest. In relation to archived material, some metadata records used in archives can record kinship information, albeit in a very granular and unconnected way, such as a free text description. However, for the kinship data to be stored in such a way in an archive, this lack of structure will limit the ability to retrieve data based on the social structures. For instance, when two metadata records refer to an individual, it might not be clear in retrospect if one individual is referred to or two individuals with the same or similar names.

Many of the existing kinship applications force a culturally specific structure onto the data being collected. This cultural specificity is also reflected in the main file format used in many genealogy applications, the Genealogical Data Communication (GEDCOM) format. This format has a rich syntax for the cultural features specific to the authors of the format (The Church of Jesus Christ of Latter-day Saints). However, it can be limiting for other social structures. One example is gender, for which the GEDCOM format allows only ‘male’, ‘female’ and ‘undetermined’ (Family History Department 1999, 61). There are many examples of societies with more than two gender categories, with genders like fa'afafine in Samoa (Macpherson 2001) and hijra in India (Sharma, 1989; Talwar 1999). Another example is that ‘unions’ in GEDCOM can at most represent the union of one female and one male spouse (Family History Department 1999, 25), whereas many other types of relationships exist, even in Western societies, such as polygamy, polyamory, same-sex relationships, and sperm/egg donors.1 These restrictions limit what can reliably be recorded, and do not offer an unbiased way to represent the variety of unions possible. If a culture being documented has any features that cannot be described in the software used, then these features will either not be recorded or a non-standard record must be

---

1 While the ASSO tag could be used to create other union types, it is intended only as an ‘indicator to link friends, neighbours, relatives, or associates of an individual’ (Family History Department, 1999, 84).
created. This will result in missing or non-interoperable records, leading to an avoidable modulation in the data collected for that culture.

Data structure and data entry

In this section we discuss the flexibility of the data structure and relations used in KinOath and the methods used to keep the resulting kinship data interoperable with other datasets. We also discuss the underlying technologies and some methods of working with the data in KinOath.

KinOath, kinship and social relations

KinOath is designed to be flexible and culturally nonspecific. This means that the data fields, such as name, date or clan, for each entity or individual are flexible, and, if required, customisations can be defined in an XML schema file. For most users the default settings will be quite adequate. However, when needed, this flexible metadata can be defined and then processed by the application via custom configurations by the user. These configurations determine, for instance, the symbol used on the diagram to represent a given gender. All of these customisations are stored in the diagram and/or in the project configuration. In order to keep the meaning of this flexible data clear, each custom field can be linked to an entry in the Data Category Registry (Broeder et al. 2010), which semantically links concepts in such a way that data can be shared across disciplines. One of the intentions of this is that data collected in different specialisations can be aggregated or shared across different fields of research, despite specialised terminologies being used, while at the same time providing a flexible structure from which specialised kinship data and diagrams can be constructed.

Underlying technology

The internal data used by KinOath is a directed graph with the primary data stored in XML files, with each entity having a separate XML file as its record. Currently this data is accessed via XQuery\(^2\) from a BaseX
Figure 4.1 Example default fields and custom fields. Left: The default fields for a person entity. Right: Adding a new field to an entity without a profile.

database. All graphics are initially generated in SVG format and this is also used as the diagram storage format.

Custom data fields

By default, person entities have the fields ‘Name’, ‘DateOfBirth’, ‘DateOfDeath’ and ‘Gender’. These fields for each entity can be customised, either by simply adding fields as needed (Figure 4.1) or by using the Clarin Component Registry to create a profile and adding it to the project settings (Figure 4.2). This means that you can specify the data fields that you need for your project. When using a Clarin profile the meaning of each field such as ‘name’ or ‘clan’ can also be defined via the ISOcat Data Category Registry. This registry provides a way to disambiguate the intended meaning of each field. It may also clarify when field names are equivalent; for instance, one dataset could use ‘human’ while another uses ‘person’, and in a given context these meanings might be equivalent. Within a Clarin profile it is also possible to de-

---

2 http://www.w3.org/TR/xquery-30.  
3 http://docs.basex.org.  
4 http://www.w3.org/Graphics/SVG.  
5 ‘CLARIN is a large-scale pan-European collaborative effort to create, coordinate and make language resources and technology available and readily useable.’ http://www.clarin.eu/content/about-clarin.  
6 http://catalog.clarin.eu/ds/ComponentRegistry.  
7 http://www.isocat.org/.
fine controlled vocabularies for each field, for instance, 'gender', which might have simply 'male' and 'female' or any number of entries as required.

**Custom relation types**

Representing kinship systems in tree-based genealogies will necessarily exclude many aspects of kinship (Bouquet 1996). There are many different relation types other than genealogical ones that need to be represented to cover the various facets of a kinship system. For this reason the relation types in KinOath can also be customised to suit the needs of the system being recorded. This can be done in the ‘diagram settings’ under ‘relation type definitions’ in the settings panel of the diagram. These customisations are stored in the diagram but can also be saved in the default diagram for a project. When defining a custom relation, the custom name, type and display style can to be entered. This new relation type will then be available to create relations between entities. The following are two examples: suckling relation (Giladi 1999), as exemplified by Figure 4.3; and song inheritance, as exemplified by Figure 4.4.
### Figure 4.3 Example custom relation configuration.

**Top:** Definition of a custom relation in the diagram settings. **Bottom left:** Example of creating a relation via the relation handles. **Bottom right:** Resulting diagram showing the custom relation in place.

<table>
<thead>
<tr>
<th>Custom Name</th>
<th>Data Category</th>
<th>Relation Type</th>
<th>Line Colour</th>
<th>Line Width</th>
<th>Line/Dash</th>
<th>Curve Line Orientation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SucklingRelation</td>
<td>other</td>
<td>other</td>
<td>2</td>
<td>0</td>
<td>horizontal</td>
<td></td>
</tr>
</tbody>
</table>

### Song Inheritance Example

**Figure 4.4** Example custom directional relations used to represent song inheritance.
Figure 4.5 Example of merging selected individuals. Top left: Initial diagram with two individuals selected. Right: Context menu for merging the selected entities. Centre left: Verification message shown before merging. Bottom left: Resulting diagram with the two individuals merged with all relations maintained.

**Merging and separating entities**

When collecting data, it can be useful to merge two entities into one (Figure 4.5), or to separate one entity into two (Figure 4.6). For example, in the case of field data of genetic relations, it may be years until it is realised that two records are of the same person, or that one record refers to two separate people. In this example, it would be necessary to select two records and merge them into one, or to expand one record out into two distinct records. KinOath keeps a journal of changes made to the project data by the user, which could be used to trace back to when a change was made. Merging or duplicating records is not stored
by the current version of the application, but this may be added in the future.

**Kin type strings and diagram types**

During the needs assessment phase of KinOath, it was determined that kin type strings can be used extensively throughout the application. One example usage of kin type strings in this application is the ability to quickly construct freeform kin diagrams by entering the kin type strings that describe the required diagram. Another is using kin type string queries to search for related individuals within a dataset. Creating diagrams from queries and generating freeform diagrams are core features of KinOath. However, these features would be of limited use if the resulting diagram could not be published. Hence, the ability to create diagrams of publishable quality is also a core feature of this application. All diagram types can be exported to JPG, PDF and SVG. In the case of SVG, the information required to generate the diagram, such as customisations and queries, can also be saved within the file, making it self-contained. If the SVG diagram contains the queries and other information required to generate the diagram, it becomes easier to replicate the results shown on the diagram, which may be crucial to support any published claims.

In the following sections we look at what kin type strings are and how they are used in this application. We also look at the various diagram types available in KinOath and the way each of them can be used.

**Kin type strings**

Kin type strings are notations used to describe kinship relations. For example, with ‘E’ signifying the individual to which the relation pertains (ego), ‘M’ signifying ‘mother’, and ‘F’ signifying ‘father’, then the relation ‘maternal grandfather of ego’ can be represented by ‘EMF’ (i.e., Ego’s Mother’s Father). KinOath offers, by default, variations on notations of the same relations. For instance, ‘mother’ can be notated as either ‘M’ or ‘Mo’. In addition to the default notations, the user can also
add, delete, or change the notations used to suit his or her needs. Kin type strings are used throughout the application to search kinship data and to generate diagrams. These kin type strings can be edited in the diagram settings (see Figure 4.7) and each kin type can use any string, any relation type and any symbol. These kin types are defined and stored in each diagram file but they can also be stored in the default diagram of a project.
Freeform diagrams

Freeform diagrams have no kinship data records and all the data required is stored in a self-contained diagram. The kin type strings or kin term definitions entered into the diagram are the only source data for these diagrams. These freeform diagrams can be created by simply entering kin type strings, with the resulting diagram showing all the entities and relations described in that string. The intention with this functionality is for the user to be able to quickly generate a diagram, which correctly describes the relations they have in mind. This supports the construction of diagrams containing matrimonial rings and specifying names and dates for individuals. This provides a very interactive way to learn how to use kin type strings or how to visualise kinship relations on a diagram. When eliciting data from a consultant, this provides a way to quickly annotate the kin structure described.
Figure 4.8 Application layout when editing a freeform diagram.

Figure 4.9 Three examples of freeform diagrams.
<KinType>:<id>;<label>;<label...>;<DOB>-<DOD>:<KinType...>

EM:#1:Jane;1721-1803:

Figure 4.10 The syntax used to generate a freeform diagram. Top: Generalised syntax showing all the optional elements. Middle: Syntax used. Bottom: Resulting diagram.

**Project diagrams**

Projects are different from freeform diagrams in that each project has its own database of entities, which can be used by any number of diagrams. Project diagrams display and query kinship records from a given project (Figure 4.11). There is a data file for each entity, which can, for example, be a person, thing, place, affiliation, kin term or event. This database can be queried to find, for example, people born after a given date or affiliations of a given type, and the result can then be shown on the diagram. Data can be imported into these projects from GEDCOM or CSV and exported to CSV. In a project diagram the changes to the kin data of a project are reflected on all diagrams using that project. This is intentionally like a database-driven drawing program.
4 KinOath Kinship Archiver

Figure 4.11 Application layout when editing a project diagram.

Figure 4.12 Adding an entity via the context menu and editing the name and gender.
Figure 4.13 Creating relations by dragging the handles from one entity to another. Left: Selecting the relation handle (dotted) on the first entity (top = ancestor, bottom = descendant). Centre: Dragging to the second entity. Right: The resulting relation.

In a project diagram new entities are added by right clicking on the diagram and selecting the type required from the ‘add’ submenu (Figure 4.12). Once added, details can be entered such as name, date of birth, gender etc. The default fields can be used or custom fields added as required.

Relations can be created either by selecting a number of entities and choosing the relation type via the context menu (Figures 4.13 and 4.14), or by dragging the dots from one individual to another. There will be a coloured dot for each relation type available.

**Kin terms diagrams**

Having an effective way to visualise the kin terms used in a language can help to understand and document their structure. This application currently provides two methods to do this. The first way is via a kin terms diagram where each kin term is listed in a table with the kin type string from ego (e.g., paternal grandmother EFM). This will generate a freeform diagram from the listed kin terms (Figure 4.15). This table of kin terms can currently only be overlayed on a freeform diagram; however, it is planned to extend this to project diagrams in future releases. The kin terms can be organised in groups and can be imported and exported to CSV format. There are a number of example kin term diagrams available in the application. The second way is to use a project diagram and create an entity for each kin term with directed relations.
to the ego and referent (Figure 4.16). This method links kin terms to specific individuals rather than the former method, which shows terms in relation to a common ego.

There are cases where ego and referent is inadequate; for instance, triangular kin terms – these are kin terms which, in order to be properly described, require a ‘propositus’ (or ‘anchor’) to be defined in addition to defining the referent and ego (Evans 2003, 34). One such example from Gundjeihmi is illustrated in Figure 4.17. These triangular kin terms can be entered using the second method described, on a project diagram with custom relations linking the relevant individuals to the kin term.
Figure 4.15 Example freeform kin term diagram with the kin terms entered in the table on the right in two groups, referential and vocative.

Figure 4.16 Example kin term when used in a project diagram.

Searching data in KinOath

There are a number of ways to query the data in a KinOath project. In this section we cover simple searches that can be done, for instance, to find individuals, and more complex searches that span relations between individuals or other entities in the project.

Kin type string queries

The kinship data stored in projects can be queried by simple keyword searches and the results then selectively added to a kin diagram. A more powerful search can be done via the kin type strings; for instance, to
al-garrng, “the one who is your mother and my daughter, given that I am your mother’s mother”, in the Gundjeihmi language (Evans, 2003: 34)

Figure 4.17 Example triangular kin term when used in a project diagram.

find an individual matching a textual query (Figures 4.18 and 4.19), then based on kin type strings select relations of that individual that match another textual query. This makes it possible to query, for instance, all individuals who are trilingual and whose great-grandparents lived in mountainous regions (providing that information has been collected and recorded with links to relevant archive metadata/data). Multiple queries can be used per kin type and each condition in the query can use the following comparators: = contains; == exact match; > greater than; < less than.

**Search panel**

Project diagrams also have a search panel (Figure 4.20), which takes a free text search string and shows the results in a browsable tree. This
Figure 4.18 Example of kin type string queries. Top: Resulting diagram. Bottom left: Query used. Bottom right: Data of one individual selected by the query.

Figure 4.19 Example of kin type string queries with multiple comparators. Top: Resulting diagram. Bottom: Query used that matches by date of birth and title.
search type uses fuzzy matching so that similar or misspelt names can be found. These search results can be browsed hierarchically in the tree so that immediate relations can be seen. If ‘graph selection’ is enabled, the selected results will be shown on the diagram, and enabling ‘expand selection’ will also show relations matching the kin type string entered. This allows the results of the search to be inspected on the diagram in context.

If the data contains time data such as event dates, this can be used to filter the information shown on the diagram. Currently this can only be done based on entities. However, it is intended to extend this functionality to also filter on specific relations. In Figure 4.21 there are three individuals, one social group and three events causing affiliation between the social group and the individuals. The events are dated 1701, 1705 and 1791. The state of this relation network can then be shown for various dates by attaching the group and the individuals to the diagram; for instance, by dragging them from the project tree onto the diagram and then querying the events by date.

Export, import, plugins and interoperability

This section covers interoperability in the context of import/export, importing updates to previously imported data, the plugin structure that is available for extending functionality and the export of data in order to perform statistical analysis.

*Interoperability*

Data can be imported from CSV, TIP\(^8\) and GEDCOM formats. Data can be exported to CSV but not to GEDCOM, because the constraints of the GEDCOM format limit the social structures that can be recorded without simplification. If the user has chosen to use the Data Category Registry entries for each field, the resulting kinship data will be more accessible to other software such as for aggregated searches.

\(^8\) http://kintip.net/kinship-network-formats-topmenu-54.
Figure 4.20 Example of search panel results. Left: Search tree showing results with browsable relations. Right: Selected result expanded by kin type ‘P’ shown in the diagram.

Figure 4.21 Example time-based query.
Figure 4.22 Example of updating existing data via a CSV import. Top left: Individuals to be updated in the project tree. Top right: Individuals to be updated on the diagram. Centre: Data of the individuals before the update (see Appendix B). Bottom: Resulting data for the individuals after the update via CSV (see Appendix C).

Importing updates

When a project needs to be synchronised with an external data source, it can be updated via a CSV import, providing the ID fields are the same for each individual in both the update and original CSV import files (Figure 4.22). This could also be used to insert columns generated via R³ or other external processing methods or for adding new information when it becomes available from a third party such as lab results.

Plugins

KinOath has been designed so that plugins can be written for specific tasks that are outside the scope of the main application. There is an existing import-export plugin, which could be extended if other formats are needed. This provides potential for extended features, such as custom graph sorters or alternative data sources, to be written without needing to modify the main application. This is an area for future development and potential collaboration.

---

3 http://www.r-project.org.
Statistical analysis

Being able to do statistical analysis on corpora based on the social and genealogical relations is potentially one of the most fruitful results of integrating archived data/metadata with kinship data. Because there are already good tools for doing statistics, KinOath focuses on integration with these tools rather than reinventing them. Integration with R and SPSS\(^\text{10}\) is achieved in the desktop application by exporting to appropriate CSV and TXT files that can be used as data sources. This data can be used in R for instance (Figure 4.23) with the standard ‘read.table()’ which results in a data frame that can be further processed and analysed.

Interoperability and archiving

In the following section we discuss the topic of linking resources and archived material, followed by the potential use of kinship data to search archived material.

Links to external resources

It is possible to link external resources to entities in a project. This can be done within the application or during the import of data. There are also archive resource linking features in the application, which provide some integration with Arbil,\(^\text{11}\) which is a metadata management application. While the integration with Arbil was added early on in the development of KinOath, this integration is incomplete and will need further work in the future. Features of the application used to create external links to IMDI metadata\(^\text{12}\) resources have been held back in the current stable version due to the need for a technical agreement on how to persistently link kinship individuals to the IMDI format. However,

\(^{10}\) http://www-01.ibm.com/software/analytics/spss.
\(^{11}\) https://tla.mpi.nl/tools/tla-tools/arbil.
\(^{12}\) https://tla.mpi.nl/imdi-metadata.
Figure 4.23 Example of processing exported data in R based on the haemophilia in European royalty. Top: Example extract of exported data for use in R. Bottom: Example R script to process the data exported from KinOath.

despite the limitations to creating external links to resources, once created they will be maintained when importing and exporting data.

Utilising external links

Currently external and resource links can only be viewed in the table when working in KinOath (Figure 4.24). Alternatively, when a diagram is saved as an SVG and viewed in a web browser, these links will be available as hyperlinks. For the intended potential of these links to be available, more work needs to be done within KinOath to better make
use of these resources, and the kinship data produced in KinOath needs to be supported by archiving services. KinOath was originally intended to have a web interface for the purpose of providing an archive search facility. However, this web application has not been completed at this stage. The issues of privacy and access to both the data and kinship records will need to be adequately addressed by the archive when such search facilities are completed. As a desktop application, KinOath does not apply permissions to the data in a project, but, if required, fields can be defined to indicate what permissions would need to be applied once the data is archived.

Visualisation and graphics

This section discusses the graphics format used in KinOath and the possible customisations that can be made.

Graphics format

The diagrams produced by KinOath use the SVG (scalable vector graphics) format and other formats, such as PDF and JPG, can be exported. Prior to publication, these diagram files can also be edited in other graphics programs, such as InkScape, Gimp, Illustrator or Photoshop, for instance.

Custom symbols

In order for the application to be truly flexible, it is necessary that custom symbols can be added and assigned to individuals and entities on the graph (Figure 4.25). This includes, but is not exclusive to, symbols indicating the gender or caste of an individual. There are a number of symbols provided in the application by default, but users can manually edit the SVG to add more specific symbols if required (Figure 4.28). These symbols can consist of graphical and/or textual elements. Once a custom symbol is defined in the diagram it can then be assigned based on features in the data for an individual. In addition to these base symbols, an overlay symbol can be applied in the same manner. For in-
Figure 4.24 Example of external links in data imported from a GEDCOM file (see Appendix A). Top: External links shown in the project tree. Bottom: External links shown on the diagram.

stance, when a metadata record contains a date of death, an overlay of a strike-through symbol is appended to the symbol of that individual.

The default symbols in this application are circle for female, triangle for male and square for unknown. Another common use of symbols is square for male and circles for females. This is easily configured in the diagram settings.

The overlay symbols could be used, for instance, to indicate known data on an individual. For instance, states like affected, unaffected, un-
known, asymptomatic, obligatory or adopted can be indicated by the default markers or custom markers could be manually added (Figure 4.27).

Figure 4.25 Example custom symbol configuration. Left: The top line shows 9 of the 30 symbols currently available by default. The second line shows an example of overlay. Right: Assigning symbols to features.

Figure 4.26 Example configuration with square representing male and circle representing female.

Figure 4.27 Example overlays configured for affected, unaffected, unknown, asymptomatic, obligatory and adopted.
Conclusion and future directions

There have been numerous applications which have tried to meet the needs of anthropologists and other researchers who frequently study kin and other social relationships. However, there are shortcomings in either the functionality, usability or affordability of this earlier software. The development of KinOath has been a recent attempt to address some of the needs of researchers, with a flexible application that can be used in conjunction with existing specialised tools. The use of kin type strings as a query language has provided a simple way of creating quick kin diagrams and a flexible way of querying kinship data. Interoperability with existing tools has provided a path to performing complex statistical analysis and the ability to migrate data from one application to another. Linking resource files and archived material to genealogies can be done, but more work needs to be done to fully explore the potential and the issues with archiving and kinship data.
KinOath is an open source project under General Public License (GPL). The current version of the application and related software can be downloaded from: http://tla.mpi.nl/tools/tla-tools/kinoath/.

KinOath is currently maintained by Peter Withers on GitHub\textsuperscript{13} and there are many areas in which the application could be further developed as funding allows. There are currently plans to collaborate with the KinSources\textsuperscript{14} and PUCK\textsuperscript{15} projects, which will involve sharing code and functionality between these projects. KinOath has been developed with a plugin structure intended to allow third-party developers to provide extended functionality to KinOath and for the functionality of KinOath to be provided to other applications. This modularity could allow for custom graph sorters, importers, exporters, symbol design and data processing. Another benefit of this structure is that new features do not need to have such an impact on the core application's stability. There is also potential for a subset of features to be provided as an application for web, mobile and tablet use, particularly in the case of crowdsourcing kinship data. KinOath already offers a lot of functionality and flexibility. However, for the full potential of this application and its use in archival searches, there will need to be further work done. Despite this, it still operates as a stand-alone application for kinship data collection and exploration.

Works cited


\textsuperscript{13} https://github.com/KinshipSoftware/KinOathKinshipArchiver.
\textsuperscript{14} https://www.kinsources.net.
\textsuperscript{15} http://kintip.net.


Appendix A: Example file imported to produce Figure 24

```
example.ged
0 HEAD
1 NAME House of Habsburg
0 @I1@ INDI
1 NAME Rudolf II
1 SEX M
1 TITL Holy Roman Emperor
1 BIRT
2 DATE 19 JUL 1552
2 PLAC Vienna, Austria
1 DEAT
2 DATE 20 Jan 1612
1 BURI
2 PLAC St. Vitus Cathedral, Prague
2 PLAC Prague, Bohemia
1 FAMC @F1@
1 OBJE
2 FILE http://en.wikipedia.org/wiki/Rudolf_II,_Holy_Roman_Emperor
```
Appendix B: Example file imported to produce the first part of Figure 22

import.csv
ID,Name,Gender,DateOfBirth,DateOfDeath,Mother,Father
315791,a,Female,,
315792,b,Male,,
315793,c,,,,315791,315792

Appendix C: Example file imported after the file in Appendix B to produce the second part of Figure 22

update.csv
ID,BirthPlace,MarriagePlace,DeathPlace
315791,48.856667 2.351042,46 4.833333,45.766667 4.833333
315792,43.330833 4.845556,45.766667 4.833333,45.766667 4.833333
315793,45.766667 4.833333,,,