Chao Pang, Computational methods for data discovery, harmonization and integration. Thesis, University of Groningen, with summary in English and Dutch.

The research presented in this thesis was mainly performed at the Genomics Coordination Center, Department of Genetics and Department of Epidemiology, University Medical Center Groningen, University of Groningen, Groningen, the Netherlands. The work in this thesis was financially supported by European Union Seventh Framework Programme (FP7/2007-2013) research projects BioSHaRE-EU (261433), PANACEA (222936) and RD-Connect (305444) and H2020 Programme research project CORBEL (654248), BBMRI-NL, a research infrastructure financed by the Dutch government (NWO 184.021.007), and NWO VIDI grant number 917.164.455.

Cover design and layout by Ridderprint. The front cover features an image of data universe, which represents the idea of data integration and discovery. The image is purchased from http://www.shutterstock.com/ with the standard license.

Printed by: Ridderprint BV | www.ridderprint.nl.

ISBN: 978-94-034-0822-4 ISBN (electronic version): 978-94-034-0821-7

university of
groningen

Computational methods for data discovery, harmonization and integration

Using lexical and semantic matching with an application to biobanking phenotypes

PhD thesis

to obtain the degree of PhD at the
University of Groningen
on the authority of the
Rector Magnificus Prof. E. Sterken
and in accordance with
the decision by the College of Deans.

This thesis will be defended in public on

Tuesday 3 July 2018 at 09.00 hours

by

Chao Pang

born on 7 May 1987
in Beijing, China

Supervisors
Prof. M.A. Swertz
Prof. J.L. Hillege

Assessment Committee
Prof. E.O. de Brock
Prof. A.L.A.J. Dekker
Prof. B. Mons

Paranymphs
D. Hendriksen
F. van Dijk

Table of Contents

Chapter 1

Introduction

Background

Biobanks and patient registries provide essential human subject data for biomedical research and the translation of these research findings into healthcare. Research interest has expanded in recent years from an interest in simple traits to a focus on complex multifactorial disorders where many genetic and environmental factors need to be taken into consideration to understand the underlying mechanism of development of diseases [1]. This requires large cohort and sample sizes and the ability to study multiple large population biobanks (for reference) and patient biobanks (for disease endpoints) in unison.

A biobank is typically defined as a collection of bio-samples and the associated human subject data collected from questionnaires and molecular experiments. The profile of the typical biobank has changed in the past thirty years from primarily small university-based patient repositories to large government-supported population-based biobanks that collect many types of data and samples [2]. The exact number of biobanks world-wide is unknown, but there are more than 200 in the Netherlands [3] and 500 in Europe [4]. Nor are these numerous biobanks small in size. For example, the largest Dutch biobank, the LifeLines biobank and cohort study, was started by the University Medical Centre Groningen, the Netherlands. Since 2006, it has recruited 167,729 participants from the northern region of the Netherlands [5] and included more than 1000 data elements covering medical history, psychosocial characteristics, lifestyle, genomic data and more.

Even with these larger biobanks, most studies still need to use data from multiple biobanks, mostly driven by their need to reach sufficient statistical power in the case of complex diseases where many small contributing factors add up to disease risk or to reach statistically sufficient numbers of patients in the case of rare diseases or phenotypes with low prevalence. One example of how use of date from multiple biobanks can increase statistical power is the Healthy Obese Project (HOP) [6]. HOP aimed at achieving a better understanding of two issues: 1) approximately 10-30% of obese individuals are metabolically healthy and 2) healthy obesity is assumed to be associated with lower risk of cardiovascular disease and mortality. Although only 2% of the total population falls under the category “healthy obesity”, HOP researchers were able to combine data from 10 biobanks to obtain 163,517 individuals with data on 100 data elements, thereby, including enough valid cases (3,387) to carry out their analysis with sufficient power.

Barriers to biobank data reuse

A major barrier to carrying out large integrated biobank studies is that biobanks are often designed independently of each other resulting in heterogeneous data that needs to be “harmonized” before integrated analysis is possible [7]. This integration is difficult to achieve and very time intensive. Fortier *et al* [8], for example, reported that only 38% of data elements could be harmonized in their study integrating 53 studies across 14 countries for a selection of 148 core data elements. Furthermore, their study took them three years to achieve, with each data element taking an average of four hours of expert input per source biobank (private communication). Their study is representative of the many research questions for which, although many suitable biobank datasets are available, it remains a huge challenge to reuse these valuable datasets. Anecdotal evidence from our years of working in the biobank community (most specifically BBMRI-NL) suggests that biobank utilization is much lower than one would expect, in large part because of the many months of menial handwork PhD students and postdocs need to spend to *discover*, *harmonize* and finally *integrate* biobank data before the actual research work can start. Each of these three barriers is detailed below:

Data discovery

Researchers conducting analyses are usually the ones who are collecting the data. Discovering which useful biobank datasets are available to reuse for a particular study is therefore the first barrier. What often happens is that researchers hear about or stumble upon a dataset in the scientific literature that could be potentially useful for their research [9]. Tracking down datasets advertised in literature, in repositories and on the Internet can be a lot of work to do due to the lack of uniform data cataloguing standards and documentation. Moreover, once biobank data have been found and integrated, they don’t always turn out to be useful for the research and thus wasting valuable researcher time. Some projects including BBMRI and Maelstrom have developed IT infrastructures [4] to integrate data descriptions from different locations based on an agreed minimal information model [10] so that researchers can access and search data through one web portal rather than having to comb the literature for the information. However, this type of approach is still limited by the level of detail that can be searched for, typically preventing researchers from discovering data with more fine-grained queries. For example, it is usually not possible to get an overview of all data elements available (counterexample: lifelines catalogue https://catalogue.lifelines.nl/) or to query for the number of individual samples having particular properties matching your research needs (counterexample: PALGA public database http://www.palgaopenbaredatabank.nl/).

Data harmonization

When suitable datasets are discovered and made accessible the next step is to make these source biobanks interoperable, a process often called “harmonization” [8]. In this process differences in data structures and data semantics need to be overcome to create a homogeneous view or “target data schema” that can be used as basis for the research. Although it is not necessary that all source biobanks use exactly the same standard procedures, tools or questionnaires for data collection, the information carried by each source needs to be inferentially equivalent. In an ideal world, information would be “prospectively harmonized”: with all new data collections reusing existing standards for data collection. Unfortunately, making this a reality would require a lot of collaboration and investment to get data owners to agree on the same data collection protocols and to rapidly produce new uniform standards for new data capture methods. Moreover, as has been said in the BioSHaRE project, "harmonizing people is more difficult than harmonizing data."

Given these difficulties, retrospective harmonization was proposed as the alternative approach by the Maelstrom Research project [11]. Retrospective harmonization consists of three steps: (i) Defining the target data schema based on the research question; (ii) Determining harmonization potential by matching biobank schemas [12]. In this step the target data elements are matched with the participating biobanks; and (iii) Defining Extract-Transform-Load (ETL) algorithms [13], i.e. developing the algorithms that take matched source data elements as inputs and converting them to the target data schema for data integration. The process is summarized in **Figure 1**.

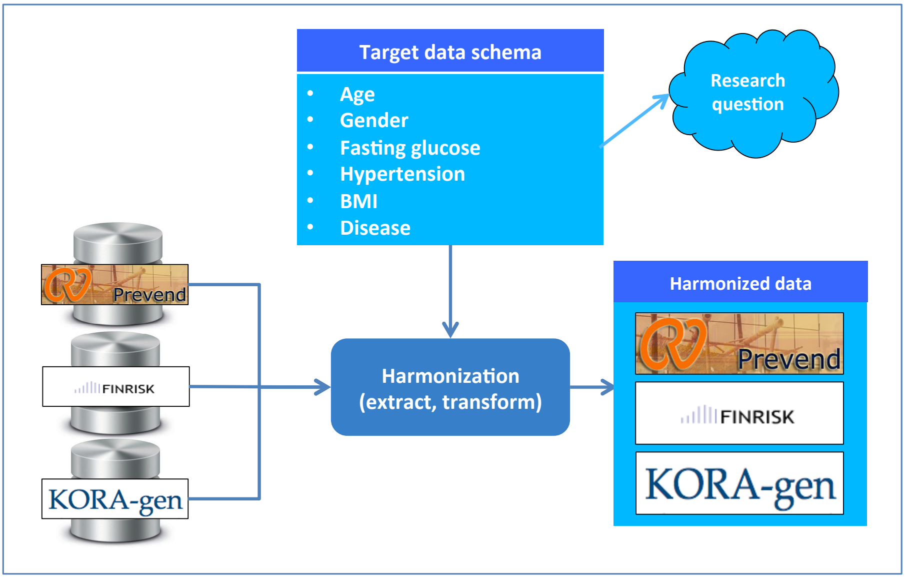


Figure 1 | Overview of retrospective harmonization. Researchers with a research question define a target data schema representing their question that consists of a group of research variables (as target data elements). Based on the target data elements, researchers try to find compatible source data elements from the participating biobanks. Values extracted from the source biobanks are transformed according to the definition of the target data schema and loaded into one harmonized dataset.

Data integration

The final barrier before the analysis can start is physical data integration. Data integration is a process to actually produce a homogeneous view of data that is derived from heterogeneous data sources [14]. There are three major data integration approaches: (i) Extract, Load and Transform (ETL) data warehousing; (ii) mediated virtual schema; and (iii) semantic integration. In ETL data warehousing, data are transformed, pooled from heterogeneous sources and loaded into a single repository. Although this approach has the advantage of responding quickly to user queries, the central repository requires frequent synchronization in order to pull the latest updates from sources. Therefore, a complementary approach has been developed called “mediated virtual schema”, in which a unified query interface is defined, and data are retrieved from sources in real time based on the mappings defined between the schemas of the central database and the data sources. This mediated virtual schema approach is more flexible due to the loose coupling between integrated data and sources but takes more time to process each query. Recently, a new type of data integration called “semantic integration” has emerged. Semantic integration focuses on the meaning of data instead of data structure, e.g. asking if by creating algorithms that can answer the questions of whether “Body Height in cm” is the same as “Length in m”? In this approach, ontologies, which are formal representations of the knowledge that describe the standard concepts and their corresponding relations in specific domains, are often used to describe the data elements and values to reduce the ambiguity.

Traditionally, the source datasets were integrated into one central database where the analysis could be carried out. However, recently, there have been many concerns about sharing data for two reasons: 1) potential exposure of sensitive individual information and 2) researchers’ concerns about losing control over valuable scientific data into which they have invested substantial time and money. To address these concerns, Amadou Gaye *et al* [15] developed a “federated” approach called DataSHIELD in which data is not centralized but rather analysis scripts are sent to each biobank hosting harmonized data. The scripts then combine the outputs back into the final result, which is returned to the user. DataSHIELD results have been mathematically shown to be equivalent to results produced by the analysis in which the individual-level data can be accessed. However, this option is often not preferred in practice because distributed analysis is methodologically and technically much more demanding.

Challenges

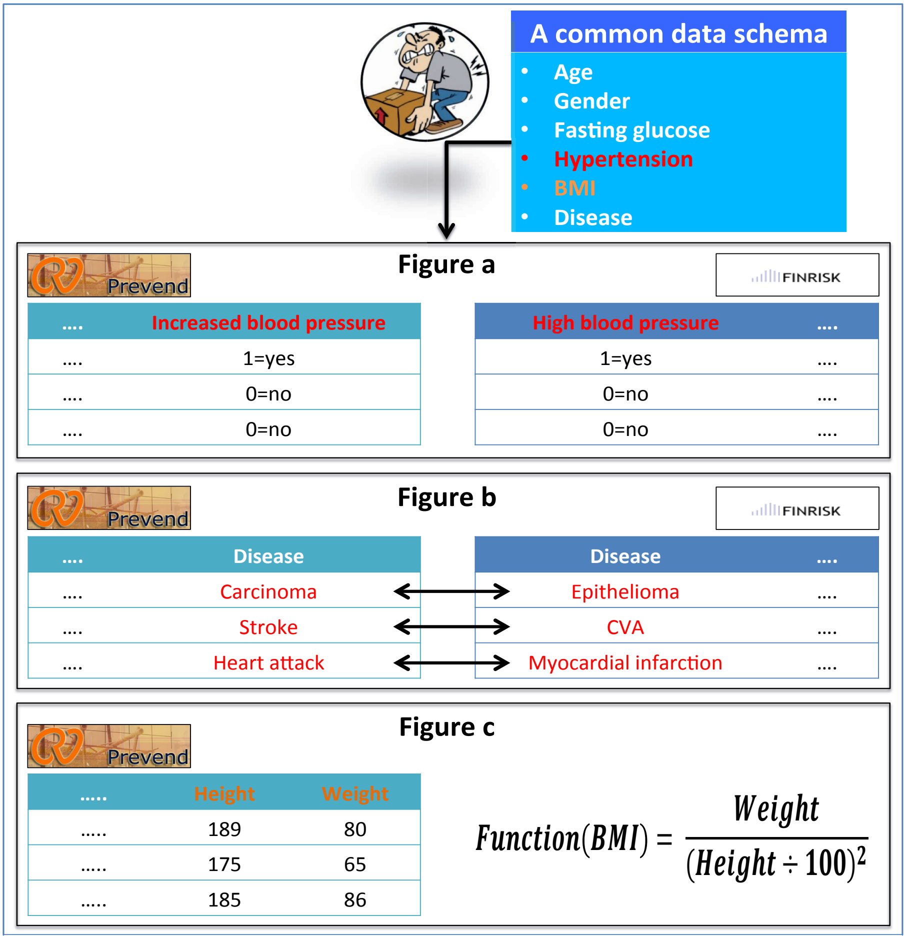
Having looked at the current patterns of biobank data reuse, we identified three major challenges that are hindering the data discovery, harmonization and integration workflow: semantic ambiguity of data definitions, non-standard coding of data values and proxy equivalent measurements.

Semantic ambiguity of data definitions

When there are multiple datasets to be matched, the data elements (column headers) are often described using different terms even though they have semantically equivalent meanings. These lexical differences between data elements (also known as “metadata”) are mainly due to (i) synonyms: multiple terms refer to the same concept, e.g. “hypertension” versus “increased blood pressure” (see **Figure 2a)**; (ii) hyponyms and hypernyms: specific terms that are instances of a more general term, e.g. “beans and peas” are instances of vegetables; and (iii) alternative definitions usually referred to as “proxy”, e.g. “Glycated hemoglobin” used as a proxy for “Blood Glucose Level” [16]. In addition there is the problem of polysemy, which is when a term has multiple meanings in different contexts. For example, “hypertensive” normally refers to a person who has high blood pressure but could also mean a drug causing an increase in blood pressure [17]. Because of these differences, matching data elements between biobanks directly based on words will not succeed. A program that can understand the meaning of those terms therefore needs to be implemented to tackle this challenge.

Non-standard coding of data values

The same ambiguity problem we saw above for metadata also occurs in the data values because people do not use standard coding systems for categorical data or - an even more complex problem - may allow free text data entry. As **Figure 2b** shows, both the Prevend and FinRisk biobanks collected information on the same disease of interest, but the two lists of diseases, while semantically the same, are lexically different. This difference creates some difficulties in integrating data from the disease column from these two biobanks because researchers would have to go through each list individually and correct each entry to the formal disease name in order to make them compatible and pool-able.



A common data schema

- Age
- Gender
- Fasting glucose
- Hypertension
- BMI
- Disease

Figure a

Prevend

....	Increased blood pressure
....	1=yes
....	0=no
....	0=no

FINRISK

High blood pressure	....
1=yes	....
0=no	....
0=no	....

Figure b

Prevend

....	Disease
....	Carcinoma
....	Stroke
....	Heart attack

FINRISK

Disease	....
Epithelioma	....
CVA	....
Myocardial infarction	....

Figure c

Prevend

.....	Height	Weight
.....	189	80
.....	175	65
.....	185	86

$$Function(BMI) = \frac{Weight}{(Height \div 100)^2}$$

Figure 2 | The three major challenges of retrospective data integration. Figure a shows an example of different terminologies used for the metadata, where the target data element “Hypertension” (highlighted in red) is described differently in two different biobanks. In the Prevend biobank it is called “Increased blood pressure” and in the FinRisk biobank “High blood pressure”. **Figure b** shows an example of different coding systems used for data values. The canonical names and synonyms are used together for describing diseases in Prevend and FinRisk, e.g. “Epithelioma” (FinRisk data value term) is actually a synonym of “Carcinoma” (Prevend data value term). **Figure c** shows an example, where the definition of the target data element (“BMI” highlighted in orange) is different from the source data elements (“Height” and “Weight” highlighted in orange). In this case we needed to create the data transformation algorithm to convert the source data values to the target.

Proxy equivalent measurements

The last challenge of integration is when researchers/biobanks use different measurements to assess what is fundamentally the same research variable. These measurements can then be used as a "proxy" of each other, see **Figure 2c**. However, because the definitions of the data values can be different, the values cannot be taken directly from the source biobank and imported into the matched target data elements. Instead, we need a transformation function or "algorithm", to convert the source data according to the definition of the target data schema [8,18–20]. Below are some examples of proxy equivalent data elements:

1. The target and source data elements are measured in different units and a unit conversion needs to take place. For example conversion of *source: Height (cm)* to *target: Height (m)*. The algorithm pseudo code in this case is target_height = source_height / 100.
2. The target and source data elements are categorical and their corresponding categories need to be matched properly. For example, *target: gender[0=male, 1=female]* versus *source: gender[1=male, 2=female]*. The pseudo code is target_gender = source_gender.map({1 : 0, 2 : 1}), by which source code 1 is mapped to target code 0 for the *male* category and source code 2 is mapped to target code 1 for the *female* category.
3. The target data element is a derived variable matched to multiple source data elements. For example, "hypertension" is the target data element described as "a person having high blood pressure" or "taking antihypertensive medications". Although the information is not available, it is possible to derive values for hypertension based on systolic and diastolic blood pressure measurements. Due to the lack of information on medications, the definition of hypertension is partially fulfilled but close enough to be used in the analysis.
4. Data structures are different across biobanks, making it necessary to combine multiple source data elements to calculate values for the target data element. For example, in the LifeLines biobank there are two source data elements "Cooked vegetables" and "Raw vegetables"

related to the target data element “frequency consumption of vegetables”, while in Mitchelstown biobank there are 10 source data elements about consumption of specific types of vegetables such as “broccoli” or “beans”. Depending on how data are collected in biobanks, algorithms need to be adjusted to combine information from all related source data elements accordingly.

Existing tools

There are a number of tools that aim to facilitate data harmonization and integration in the biomedical domain, thus what follows below is a short review of the more common systems and the extent to which they address the challenges describe above.

eleMAP

eleMAP is a harmonization and semantic integration tool that can recode metadata and data values using ontologies through the BioPortal ontology service [21]. Users first match source data elements to the ontology terms via a search box. Additionally, users need to match the allowed values to ontology terms in cases of categorical variables, e.g. the data element “Gender” is mapped to “NCI:C17357” and the allowed values “males” and “females” are mapped to “NCI:C20197” and “NCI:C16576”, respectively. Second, users can upload actual data with the same column headers that have been matched to ontology terms. Based on those matches, eleMAP is able to recode all the data values with the ontology term-identifiers in one go. While innovative, eleMAP has the following shortcomings relative to direct application in the biobanking domain: I) although it provides a search box to quickly locate the proper ontology terms, the matching process still needs to be done one-by-one, which is not very efficient especially when the target and source data schemas contain many data elements (such as the thousands of elements in biobanks); II) eleMAP does not support harmonization using local terminologies, only the ontologies available on BioPortal can be used. In practice, the target schema is usually not defined using standard ontology terms, but rather via a locally-created codes list of target data elements.

eleMAP will therefore fail to harmonize such data elements; and while III) eleMAP is convenient for harmonizing values of simple data elements, such as gender and weight (as seen in their video tutorial https://victr.vanderbilt.edu/eleMAP/icontroller.php?branch=help), it does not provide sophisticated data harmonization algorithms to handle more complex data elements, a feature which is needed to integrate proxy equivalent data elements.

ZOOMA

ZOOMA [22] is a high-performance ontology matching tool that can be used to semi-automatically annotate biological data with selected ontologies. It provides an easy-to-use graphical user interface (GUI) on a web page, and users can simply copy/paste a column of data values into the text editor, choose the ontologies of interest and push the button. ZOOMA then produces a report containing a list of potential matches from the selected ontologies based on the lexical similarities [12]. The user can download those ontology term matches in a CSV (comma separated values) file easily read by humans or parsed by computers. Most importantly, ZOOMA enables the incorporation of knowledge provided by human curators during the annotation process. ZOOMA produces two types of matches (“Automatic” or “Curation required”) based on whether or not there is manually curated knowledge that could support such suggested matches. When there is evidence present, matches are flagged as “Automatic” and don’t need any further inspection. Without any evidence, even if they are perfect matches, they are flagged as “Curation required” and therefore need curators to investigate. Although ZOOMA addresses the challenge of non-standard coding, it only provides the qualitative evidence to indicate the quality of candidate matches. In practice, users like to have quantitative evidence about match value, e.g. a similarity score ranging from 0-100%, to assist them in their selection of a final match. In addition, ZOOMA would need extensions to address semantic ambiguity of metadata and proxy-equivalent data harmonization.

SAIL

SAIL is a web application developed for managing, browsing and searching biobank samples [23]. More importantly, it provides the capability for admin users to harmonize the sample data by defining “relations” between data elements across data schemas (which they refer to as vocabularies). This includes, for example, synonymous relations and partial match relations, which is a way to link semantically similar or same data elements, e.g. “glucose level” is a partial match for “fasting glucose”. However, the harmonization work is done manually by data curators, which is feasible because SAIL is used to match data structures for biobank samples that use relatively simple standards such as MIABIS [10]. However, to match 1000s of data elements between biobanks, automatic approaches are required to support data discovery, harmonization and integration.

tranSMART

tranSMART is an open-source knowledge management and data analysis platform [24] that has incorporated the Extract, Transform and Load (ETL) data integration tools. The philosophy behind tranSMART is that researchers should focus on research rather than data processing, and therefore source data are loaded and matched to a common data model by skilled staff members in tranSMART. The common data model covers domains such as clinical trial data, SNP data and gene expression data. All loaded source data conform to the same structure and meaning, which are thus automatically compatible and pool-able. tranSMART data loading can be described into two steps. First, an experienced data analyst defines matches in a template for both source data elements and data values using global reference terminologies based on the standard practices. Second, an ETL developer runs data transformation algorithms based on the mapping template to create the data in a standard format, which will eventually be loaded into tranSMART. Detailed documentation can be found at http://transmartfoundation.org/manuals-and-tutorials/. Although tranSMART provides the complete set of ETL tools, there is one major barrier to its wider use. Only tranSMART staff members can perform data transformation as it doesn't provide automated assistance to speed up the discovery, harmonization and integration task. Thus, tranSMART might make a nice target system to host the integrated data, but it doesn't address the challenges we described above in section 1.3 (although the methods described in this thesis might be a nice add-on for tranSMART).

OPAL

OPAL [19] is a web-based database application specifically designed for managing and harmonizing biobank data that is widely used for integrated biobank studies. It accepts datasets in various formats such as Microsoft Excel, SPSS and Extensible Markup Language (XML). The core feature of OPAL is the capability to convert source data to the target data schema and combine them by allowing users to define ETL data transformation algorithms. In this process the biobank data are converted to a common standard (data schema) such that the data elements measured in individual biobanks are compatible. To do this, the OPAL development team has designed an algorithm syntax therefore called “Magma” [18], written in JavaScript programming language, which might be reusable to address the challenges in this thesis (see chapter 4). However, harmonization work still needs to be done manually in OPAL and it doesn’t provide an easy way to discover source data elements for target elements in the matching screen (where algorithms are developed). Finally, OPAL doesn’t support recoding the data values using the external coding systems or reference terminologies such as SNOMED-CT and Disease Ontology.

Summary

The tools described above address only some of the data integration challenges (see comparison in **Table 1)**, and all require much handwork. There is therefore a need for (semi-)automatic computational methods for data element discovery, recoding of data values and generation of integration algorithms.

Table 1 | **Requirements of the (semi-) automatic data integration system**

		tranSMART	SAIL	eleMAP	OPAL	ZOOMA
Semantic integration	Automatically recoding data values					Y
	Manually recoding data values	Y	Y	Y		Y
ETL data integration	Define target schemas	Y	Y	Y	Y	
	Automatically finding data elements					
	Automatically generating algorithms					
	Manually finding data elements	Y	Y	Y	Y	

This thesis

This thesis aims to overcome barriers to biobank data reuse. These barriers exist because biobanks do not apply the same standards and terminologies for data collection, and the resolution of these differences takes up much time and effort on the part of researchers. We therefore hypothesized that computational methods and tools can remove much of this handwork and assist researchers in retrospective data harmonization and standardization as basis for data discovery and integration. To evaluate this hypothesis, we researched and developed relevant computational methods and evaluated them in practical software implementations on a mission to convert any source datasets to any target data model in an automatic fashion. For this implementation we chose to use open source MOLGENIS software because it provides complete freedom in data structure and because the system is maintained at the University Medical Center Groningen, allowing us to influence its development for the purpose of this thesis.

Based on the aims and challenges, we have defined four specific research questions that are addressed in each of the chapters separately.

Question 1: Can we (semi-)automatically discover which biobank data elements match desired/standardized research variables? (e.g. "increased blood pressure" → "Hypertension"). This question mainly addresses the first challenge, semantic ambiguity, which is essential for efficiently discovering a small set of relevant data elements from a large number of all biobank data elements and for harmonizing the non-standard source data elements by linking them to the standard target data elements. In chapter 2, we discuss BiobankConnect an application that rapidly connects data elements for pooled analysis across biobanks using ontological and lexical indexing.

Question 2: Can we (semi-)automatically recode biobank data values to (standard) coding systems by matching them with the common terminologies or ontologies? This question corresponds to the second challenge, non-standard coding systems, which is about harmonizing data values (string type) by matching locally-used coding systems or free text to globally defined coding systems such as ontologies. In chapter 3, we discuss SORTA, an application for ontology-based re-coding and technical annotation of biomedical phenotype data.

Question 3: Can we (semi-)automatically generate data transformation algorithms to convert biobank source data to a common standard data schema so that researchers can obtain a large dataset to carry out their analyses? This question corresponds to the third challenge, proxy equivalent measures, which involves integrating different source datasets based on a standard target schema via data harmonization. In chapter 4, we discuss MOLGENIS/connect an application for semiautomatic integration of heterogeneous phenotype data with applications in biobanks**.**

Question 4: Can we (semi-)automatically match different standard data models so that data flow can be easily enabled among them? The last question is an extension of the question 1 (discovery of data elements), which is about discovery of the relevant biobanks at a global scale. In chapter 5, we discuss BiobankUniverse an application utilizing automatic matchmaking between datasets with features like data discovery and integration.

Chapter 2

BiobankConnect – a software to rapidly connect data elements for pooled analysis across biobanks using ontological and lexical indexing

Chao Pang,[1,3] Dennis Hendriksen,[1] Martijn Dijkstra,[1] K. Joeri van der Velde,[1,2]
Joel Kuiper,[1,3] Hans Hillege,[3] Morris Swertz[1,2]

Genomics Coordination Center, Groningen Bioinformatics Center, Department of Genetics and Department of Epidemiology, University Medical Center Groningen, University of Groningen, The Netherlands

Journal of the American Medical Informatics Association *2015, 65-75*

(November 2014)

DOI: 10.1136/amiajnl-2013-002577

Abstract

Objective Pooling data across biobanks is necessary to increase statistical power, reveal more subtle associations, and synergize the value of data sources. However, searching for desired data elements among the thousands of available elements and harmonizing differences in terminology, data collection and structure, are arduous and time-consuming.

Materials and methods To speed up biobank data pooling we developed BiobankConnect, a system to semi-automatically match desired data elements to available elements by: I) annotating the desired elements with ontology terms using BioPortal; II) automatically expanding the query for these elements with synonyms and subclass information using OntoCAT; III) automatically searching available elements for these expanded terms using Lucene lexical matching; and IV) shortlisting relevant matches sorted by matching score.

Results We evaluated BiobankConnect using human curated matches from EU-BioSHaRE, searching for 32 desired data elements in 7,461 available elements from six biobanks. We found 0.75 precision at rank 1 and 0.74 recall at rank 10 compared to a manually curated set of relevant matches. In addition, best matches chosen by BioSHaRE experts ranked first in 63.0% and in the top-10 in 98.4% of the cases, showing our system has the potential to significantly reduce manual matching work.

Conclusion BiobankConnect provides an easy user interface to significantly speed up the biobank harmonization process. It may also prove useful for other forms of biomedical data integration. All the software can be downloaded as a MOLGENIS open source app from http://www.github.com/molgenis, with a demo available at http://www.biobankconnect.org.

2.1 Introduction

Researchers increasingly need large data sets to uncover the subtle statistical associations between phenotypes and diseases. It is therefore desirable to pool data from multiple biobanks for analysis.[8] However, existing biobanks are usually designed specifically to local requirements and not to be similar to other resources. It therefore requires an incredible amount of time and effort to find relevant data elements across many different biobanks and to combine these into one statistically testable data set.[25]

The process of integrating comparable, but not necessarily identical, data from different biobanks is often referred to as 'harmonization'[8] and can be separated into several steps:[12]

I) Research question parameterization: define data elements of interest based on the research question, e.g. to statistically derive a prediction model for the risk of developing diabetes, data elements for well-known risk factors such as age, smoking status, blood pressure and cholesterol are desired.[26]

II) Schema matching: assess harmonization potential by comparing desired elements within the 'data dictionaries' of each biobank. These are usually tab-delimited files that contain all the data elements available in the biobank and their corresponding information such as name, label and definition (**Figure 1**). The challenges lie in finding the matching elements and deciding whether they are scientifically comparable enough to be used for a pooled analysis.

III) Data integration: transforming source data into the target schema by creating algorithms based on the matches produced by schema matching that can derive the values of desired data elements from each of the biobanks. For example, pooling the data element 'body mass index' from the NCDS biobank cannot be done directly because there is no such element available; the alternative elements 'height in cm' and 'weight in kg' are therefore used to calculate BMI. During the calculation, the unit for 'height in cm' is converted into a unit in meters.

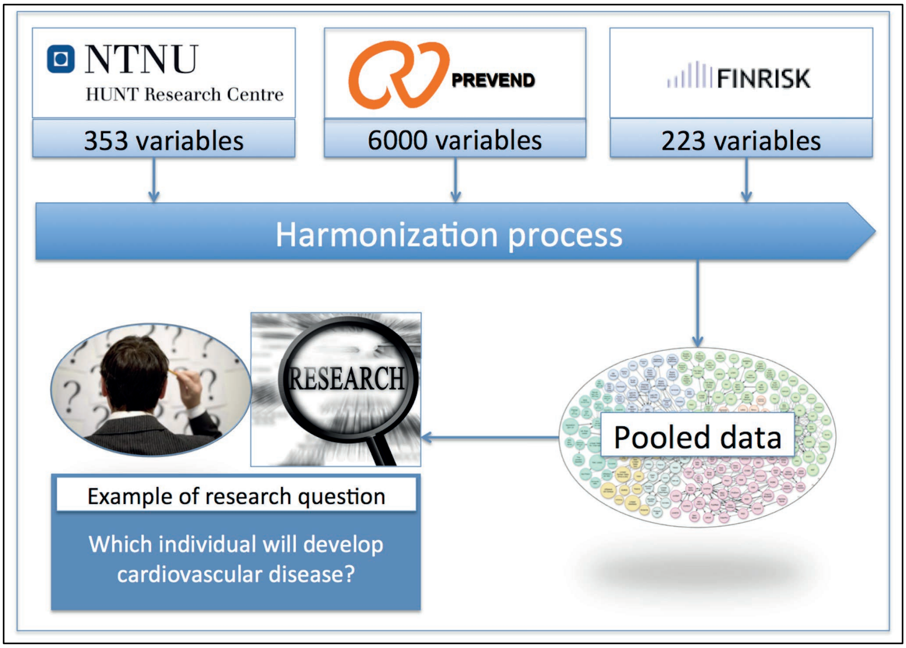


Figure 1 | **Harmonization process**. Many studies need to pool data in order to reach sufficient statistical power, however matching data elements of interest to the available data elements is a daunting task.

In this paper we describe how we can dramatically reduce the time needed for the second step — 'schema matching'. The current process is for human experts to go through all the data dictionaries manually to identify potentially matching data elements, e.g. the Prevend data dictionary contains 6,000 data elements including follow-up studies. Even when researchers are familiar with the content of a data dictionary, it takes multiple iterations to find relevant data elements and decide whether these are a complete, partial or impossible match. This requires many detailed assessments such as 'is self-reported' or 'by a physician', 'is whole life' or 'current status only', since even a small change in the way information is collected can substantially modify the scientific comparability of elements. This may take 4 hours per data element (personal communication from the BioSHaRE project).[8,25] Often the desired data element is not available and the best one can do is to identify a proxy element that is strongly related to the element of interest statistically and which can be used as an indirect measure.[26] Moreover, the type and definition of potential proxies can vary greatly across biobanks, there are usually hundreds of available data elements in each biobank, and the descriptions use different local terminologies.[8] Hypertension, for example, can also be described as ‘high blood pressure’ or ‘increase in blood pressure’.

2.2 Background

The challenge in determining harmonization potential can be generalized as matching data elements from two schemas using unstructured data element descriptions.[27] In the literature there are two major candidate methods to automate this procedure: lexical matching and semantic matching.

Lexical matching

Lexical matching is a method to measure the similarity between two strings. Prior to matching, strings need to be processed by normalization procedures such as lowering case, removing punctuation, blank characters and etc. There are two matching algorithms that are relevant:[12] I) Edit distance techniques using the minimal number of operations that needs to be applied to one string in order to get to the other one, such as N-grams and Levenshtein distance. II) Token-based distance techniques, derived from information retrieval research, e.g. Vector Space Models (VSM), which are usually recommended for matching long strings. They treat strings as bags of words, in which each dimension represents a word, with its length representing the number of occurrences of that word. Similarity can be measured using a Cosine similarity function that calculates the cosine angles between two vectors representing two different strings. Considering that the descriptions of biobank data elements are usually in the format of unstructured long strings, it was logical to choose a token-based distance matching algorithm over other approaches for our system.

Semantic matching

Semantic matching searches for correspondences using knowledge about the concepts and their relationships.[28] In ontologies, some related concepts are connected with a *subClassOf (is-a)* relationship, which construct the backbone of taxonomic structures. These concepts are considered to be quite similar and could therefore be considered a partial match. For example, matching 'Parental Diabetes Mellitus' with 'Father Diabetes' cannot be achieved using the lexical matching strategy because the relation between 'Father' and 'Parent' cannot be determined by synonyms. However in an ontology, the fact that 'Father' is a subclass of 'Parent' is stated explicitly, so matching 'Father Diabetes' with 'Parental Diabetes Mellitus' becomes possible. Other than the *is-a* relationship, the concepts could also be connected by associative relations such as *part-of* and *has-location*. However, matching based on these relations is not useful in this project.[12]

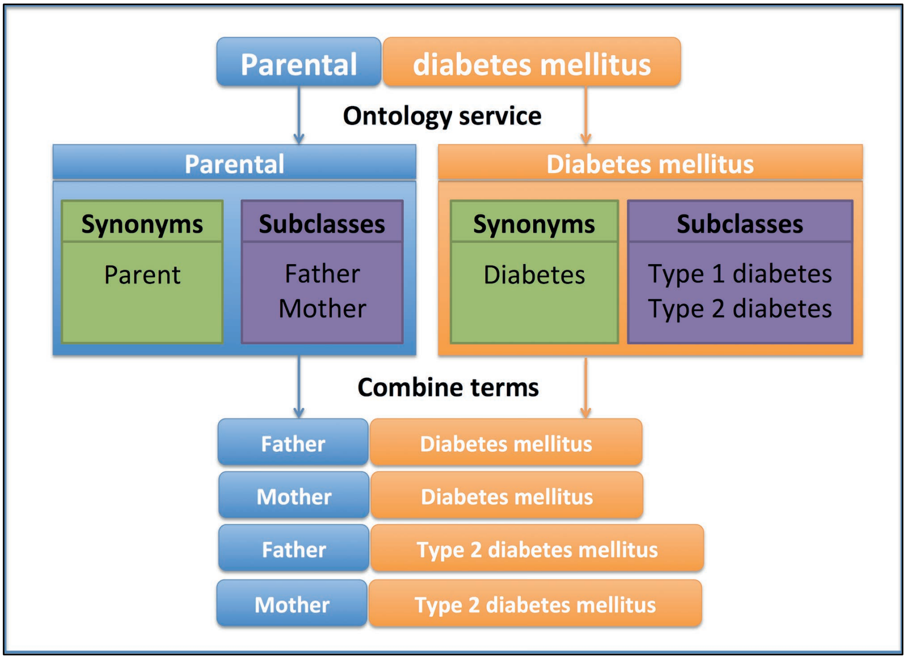


Figure 2 | **Example of query expansion**: 'Parental diabetes mellitus' is annotated with the ontology terms 'Parental' and 'Diabetes Mellitus'. Then the terms are expanded based on synonyms, resulting in 3 terms for 'Diabetes Mellitus' and 3 terms for 'Parental'. All 3x3=9 combinations are used for the search (only 4 are shown here).

Semantic matching techniques make sense of biobank schema matching because different terminologies are often used to describe equivalent concepts and/or more specialized data elements are available that can be used as a proxy for the desired concept. To find these correspondences, query expansion is a useful method to enhance the search by adding semantically similar terms to expand the original query in order to match more data elements.[29] Ontologies provide the background knowledge for such query expansion. Normally synonyms and hyponyms (subclasses) provided by the ontology are used. An example of a query expansion for ‘Parental diabetes mellitus’ is shown in **Figure 2**.

Existing tools

There are several lexical- and semantic-matching tools that could benefit our system: Díaz-Galiano et al illustrated the use of synonyms for query expansion to improve the performance of the retrieval system.[30] Each query was matched against a set of MeSH terms (concept and synonyms), and as long as the MeSH term could be found in the query, its corresponding set of terms would be appended to the query. A similar approach was used in GOPubmed, where a query was submitted to PubMed and retrieved abstracts were matched against ontology terms in Gene Ontology using a string-matching algorithm based on synonyms.[31,32] Rodriguez et al, Nilsson et al, and Voorhees et al described similar approaches using ontologies for query expansion to resolve ambiguous terms.[33–35] Not only synonyms but also hyponyms (subclasses) were extracted from ontologies and used to expand queries. The main difference between these projects was the choice of ontologies, implying that the choice depends on the data that need to be dealt with; the data therefore require careful evaluation. Finally, Aleksovski et al described a strategy in which they mapped two lists of unstructured medical terms from two hospitals in Amsterdam. Their strategy best addresses our matching problem.[27] There were two major steps in their process: I) automatically annotating two lists of terms with DICE ontology terms using a string-matching algorithm, which they called the ‘ontology term anchoring’, in order to enrich semantics for both lists, and II) automatically matching two lists that were annotated with ontology terms using existing ontology matchers such as FOAM and S-Match.[36,37]

We also searched for tools to manage biobank data dictionaries, and found the CIMI clinical information modeling initiative,[38] caDSR cancer data standards registry of common data elements,[39] and the Observ-OM phenotype system,[40] which all deal with data models not unlike the ‘data schemas' in our project. But, to our knowledge, there is still little automation support to map non-standard data to these elements, with caDSR coming closest to our needs with UML annotation tools (Semantic Integration Workbench, SIW) that used a simple search for matches by name. We decided to combine elements from these tools in BiobankConnect.

2.3 Methods

We implemented a three-step harmonization strategy: First, data elements of interest, which are defined based on the research question, are manually annotated with ontology terms, e.g. users can choose from a drop-down menu to annotate a data element of interest such as smoking status or cardiovascular disease. Then, these ontology terms are used to automatically scan the descriptions of the thousands of available data elements from each biobank to find potential matches. Finally, all candidate matches are sorted from ‘best’ to ‘worst’ so researchers can quickly decide on a useful match.

Figure 3 shows an overview of our matching strategy, which can be seen as a simplified version of Aleksovski et al.[27] The process is implemented on top of the Observ-OM data model for describing the data elements and the MOLGENIS web database software in Java.[40,41] Details of each step are described below.

Step 1. Manually annotate the search elements with ontology terms

To improve the accuracy of matching, we enable researchers to annotate data elements of interest with ontology terms either automatically or by hand. We added this option because some concepts are described in ontologies with a slightly different label than the desired data elements, something a human expert can quickly resolve. Moreover, there are typically only a few data elements of interest and this manual work is therefore limited. For example, to apply a prediction model for type 2 diabetes, about 10 predictors (data elements of interest) needed to be ontologically annotated.

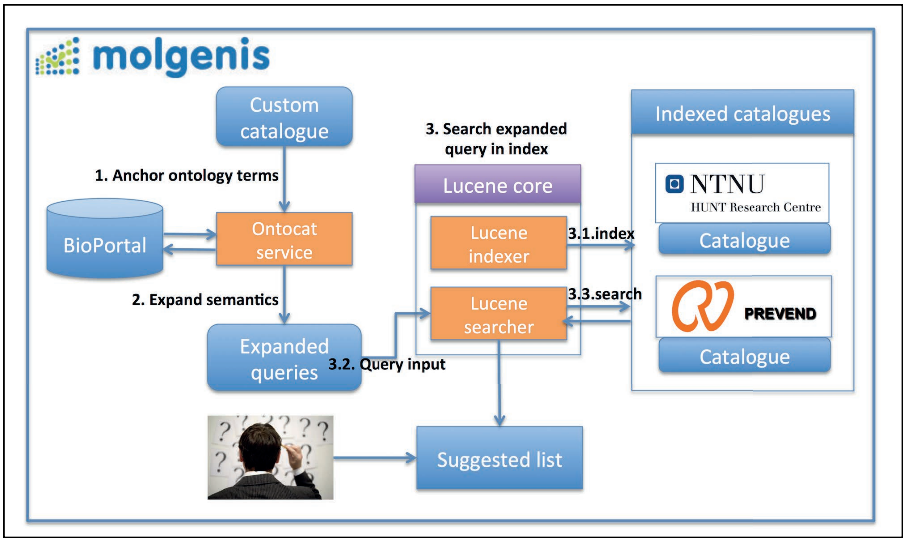


Figure 3 | **Overview of BiobankConnect**. Matching data elements of interest (target) to all the available data elements (source), based on the knowledge from the ontology terms.

To access ontologies we use BioPortal [42], an online ontology service with more than 400 ontologies currently available. To carefully select which ontology to use for our test case, we indexed questionnaires (collections of data elements) for 53 studies[8] taken from the P3G observatory and matched those against all the ontologies that are available on BioPortal.[43] Among these ontologies, we chose NCI thesaurus and SNOMED Clinical Terms (SNOMED CT) because both are characterized by broad ranges (90,000 and 300,000 concepts, respectively) and matched the most terms in the 53 studies. For medication-related data elements we use the Anatomical Therapeutic Chemical (ATC) Classification System, because ATC codes are commonly used to store information regarding medication usage. Users can use any or all of these ontologies. The ontology terms are retrieved from BioPortal[42] using the OntoCAT[44] software and stored in a local Lucene index.[45] Use of this local index solves the problem of slow response when many requests are made over slow Internet connections.

Step 2. Automatically expand semantics for search elements

The ontology annotations are used to automatically expand a query for data elements of interest using synonyms and subclasses. This succeeds in many cases where the biobank does not contain a perfectly matched data element by retrieving similar or more specific elements that can be used as proxies. For example, when matching ‘Current use of alcohol’, the annotated ontology term ‘Alcoholic beverage’ in the NCI ontology lists more specific types of alcoholic beverages, such as ‘beer’, ‘wine’ and ‘liquor’, and biobanks with data elements that are related to any of these beverages can then be matched. A complete query is created based on the expansions of the desired data element definitions using both synonyms and subclasses from the ontology terms. For example, the query ‘Hypertension' is written as {‘Hypertension’ OR ‘Increased blood pressure’ OR ‘High blood pressure’ OR ‘Hypertensive disorder’ OR ‘HTN’}. **Figure 2** shows another example. When the data elements of interest are not annotated with ontology terms, simply the labels will be used as the query in the search.

Step 3. Lexical matching of the expanded query

Finally, all data dictionaries are searched via lexical matching and potential matches are shortlisted for manual decision-making. The retrieved data elements are sorted by Lucene VSM (Vector Space Model) scores and then presented as ordered lists of candidate data elements per biobank from which users can decide on a suitable match. An all-to-all comparison of search data elements against all elements from all biobank dictionaries is a computationally expensive task, which took days in our original prototype. To speed up this process we pre-indexed all the data dictionaries using Lucene.[45] Prior to indexing, the sophisticated language pre-processing of Lucene removes ‘stop words’ (such as ‘what’ and ‘where’) from data elements to increase the sensitivity of matching. Lucene also stems terms in data elements so that different variations can be recognized during a search, for example, the stem for ‘smoking’ and ‘smoked’ is ‘smoke’.

2.4 Evaluation

To evaluate BiobankConnect we used schema matching data from the EU-BioSHaRE Healthy Obese Project (HOP).[20,46] In this project a team of biobank experts integrated a schema of 32 data elements for pooled analysis across the six biobanks with 7,461 data elements available: Prevend (NL),[47] NCDS (UK), HUNT (SE), MICROS(IT), KORA(GE), and FinRisk (FI). First, we calculated precision/recall metrics by comparing the automatically retrieved 'relevant matches' with a human curated match set created by the authors. Secondly, we evaluated the ordering of the results by assessing the ranks of the best matches that were eventually chosen for use in the pooled analysis of this "healthy obese" study.

Precision and recall

Finding relevant matches out of all possible matches has, at its base, a binary classification. Its performance can be evaluated using the widely accepted measures of precision (the fraction of retrieved instances that are relevant), and recall (also known as sensitivity, the fraction of relevant instances that are retrieved).[48]

$$Recall = \frac{\#Relevant\_matches\_found}{\#All\_relevant\_matches}$$

$$Precision = \frac{\#Relevant\_matches\_found}{\#Retrieved\_matches}$$

In order to calculate recall, we classified all possible matches between all the 32 desired and all the available data elements, and marked them as relevant or not for five of our biobanks (we excluded the largest). Out of 41,184 possible matches, 420 were classified as relevant (see **Supp Table S1 for** the full data**)**.

Prioritization of matches

While precision and recall are good performance measures, not all the relevant matches will be used for data integration. In practice, human experts will decide to use one or two data elements from the list of relevant matches for their research, e.g. out of two data elements, 'weight at baseline' and 'weight at year 1', only the first might be chosen because baseline data is preferred. Ideally, these best matches should be at the top of the list of relevant matches.

We were fortunate to have a set of 191 manually selected best matches that were used in the pooled analysis of the HOP. See Fortier et al and Dorion et al for the guidelines used for creating the matches, the qualifications of the experts, and the quality assurance procedures.[8,20] Using this set, we evaluated the prioritization of matches in the results generated by BiobankConnect (see **Supp Table S2** for the full data).

User interface

The harmonization workflow can be summarized as follows:

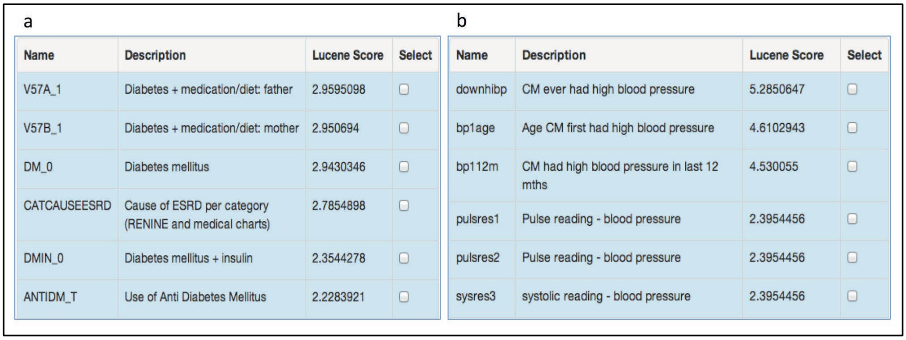
I) Upload a target data dictionary containing data elements of interest;

II) Upload one or more source data dictionaries with the data elements available from the biobanks;

III) Either manually or with the help of the annotation wizard, tag the target data dictionary with ontology terms;

IV) Choose the target data dictionary as well as the source biobank dictionaries;

V) Automatically produce the shortlist of candidate matches to choose from.



Name	Description	Lucene Score	Select
V57A_1	Diabetes + medication/diet: father	2.9595098	□
V57B_1	Diabetes + medication/diet: mother	2.950694	□
DM_0	Diabetes mellitus	2.9430346	□
CATCAUSEESRD	Cause of ESRD per category (RENINE and medical charts)	2.7854898	□
DMIN_0	Diabetes mellitus + insulin	2.3544278	□
ANTIDM_T	Use of Anti Diabetes Mellitus	2.2283921	□

Name	Description	Lucene Score	Select
downhibp	CM ever had high blood pressure	5.2850647	□
bp1age	Age CM first had high blood pressure	4.6102943	□
bp112m	CM had high blood pressure in last 12 mths	4.530055	□
pulsres1	Pulse reading - blood pressure	2.3954456	□
pulsres2	Pulse reading - blood pressure	2.3954456	□
sysres3	systolic reading - blood pressure	2.3954456	□

Figure 4 | **Matching results produced by BiobankConnect.** Panel (a) shows matching data elements for ‘Parental diabetes mellitus’ in Prevend. The gold standard matches are of two data elements, V57A_1 and V57B_1, located in the 2nd and 3rd positions. Panel (b) shows the matching data element for ‘History of Hypertension’ in the NCDS database. The best match in the experts’ opinion is ‘downhibp’, located in the 1st position on the candidate list. CM represents ‘Cohort Member’.

Figure 4 shows an example of the matched results from BiobankConnect for the search elements ‘Parental diabetes mellitus’ and ‘History of hypertension’ in the Prevend and NCDS biobanks, respectively, sorted by Lucene score from high to low. **Figure 4a** shows that ‘Parental diabetes mellitus’ was matched successfully by using the information from subclasses of ontology term annotations, e.g. ‘father’ or ‘mother’ must be a ‘parent’. **Figure 4b** shows the successful use of synonyms in matching ‘History of hypertension’ in NCDS. Note that the description ‘Ever had high blood pressure’ is quite different from the database term ‘Hypertension’, which could not be matched automatically by only using string-matching algorithms. However, with the BiobankConnect harmonization method, ‘History of hypertension’ is annotated with the ontology term ‘NCI:Hypertension’, which has a list of synonyms including ‘High blood pressure’ and using this knowledge ‘History of hypertension’ was matched with ‘CM ever had high blood pressure’ (CM: cohort member) in NCDS within seconds.

We annotated the data elements with ontology terms (without extensive training or instruction) using a rather simple approach in which as long as any synonyms of the ontology term were similar to the data element description, the ontology term would be used for annotation. For example ‘Parental diabetes mellitus’ was annotated with NCI:parent and NCI:Diabetes Mellitus; the full list of ontology terms and external knowledge annotations for all 32 data elements is given in **Supp Table S3**.

2.5 Results

Precision and recall of relevant matches

We calculated BiobankConnect`s precision and recall for 32 desired data elements across the five biobanks, with a total of 41,184 possible matches, of which 420 were classified as relevant. Overall, we observed an average precision of 0.75 at rank 1 and recall of 0.74, 0.82, 0.88 at rank 10, 20, 50 respectively (see **Table 1** and **Figure 5)**.

Table 1 | Precision/recall performance. Calculated per biobank and total.

	FinRisk		Hunt		KORA		MICROS		NCDS		Total	
Rank	P	R	P	R	P	R	P	R	P	R	P	R
1	0.91	0.50	0.61	0.16	0.88	0.53	0.73	0.27	0.59	0.17	0.75	0.28
2	0.68	0.72	0.65	0.34	0.67	0.79	0.53	0.37	0.48	0.27	0.60	0.44
3	0.57	0.88	0.59	0.46	0.48	0.83	0.45	0.46	0.37	0.30	0.49	0.52
4	0.45	0.90	0.53	0.55	0.40	0.89	0.39	0.52	0.31	0.33	0.42	0.58
5	0.39	0.95	0.47	0.60	0.34	0.92	0.33	0.56	0.27	0.36	0.36	0.62
6	0.34	0.97	0.42	0.64	0.31	0.96	0.30	0.61	0.25	0.39	0.32	0.65
7	0.29	0.97	0.39	0.69	0.27	0.96	0.27	0.63	0.23	0.41	0.29	0.68
8	0.26	0.97	0.37	0.73	0.25	0.98	0.25	0.67	0.21	0.44	0.27	0.71
9	0.23	0.97	0.35	0.77	0.24	1.00	0.24	0.68	0.19	0.44	0.25	0.72
10	0.22	0.98	0.33	0.81	0.22	1.00	0.22	0.70	0.17	0.44	0.23	0.74
11	0.20	0.98	0.31	0.82	0.21	1.00	0.21	0.71	0.16	0.44	0.22	0.75
12	0.19	0.98	0.29	0.83	0.20	1.00	0.20	0.72	0.15	0.45	0.21	0.75
13	0.18	0.98	0.27	0.84	0.19	1.00	0.19	0.74	0.14	0.46	0.20	0.76
14	0.17	0.98	0.25	0.84	0.18	1.00	0.19	0.77	0.14	0.47	0.19	0.77
15	0.16	0.98	0.24	0.85	0.17	1.00	0.19	0.79	0.13	0.49	0.18	0.78
16	0.15	0.98	0.23	0.86	0.16	1.00	0.18	0.82	0.13	0.50	0.17	0.80
17	0.14	0.98	0.22	0.86	0.16	1.00	0.18	0.84	0.13	0.51	0.17	0.79
18	0.14	0.98	0.21	0.87	0.15	1.00	0.18	0.85	0.12	0.51	0.15	0.81
19	0.13	0.98	0.20	0.87	0.14	1.00	0.17	0.87	0.12	0.52	0.16	0.81
20	0.13	0.98	0.19	0.88	0.14	1.00	0.17	0.87	0.11	0.53	0.14	0.82
30	0.09	0.98	0.13	0.91	0.11	1.00	0.14	0.93	0.08	0.57	0.11	0.85
50	0.06	0.98	0.09	0.94	0.10	1.00	0.11	0.96	0.06	0.64	0.08	0.88

P Precision; R Recall.

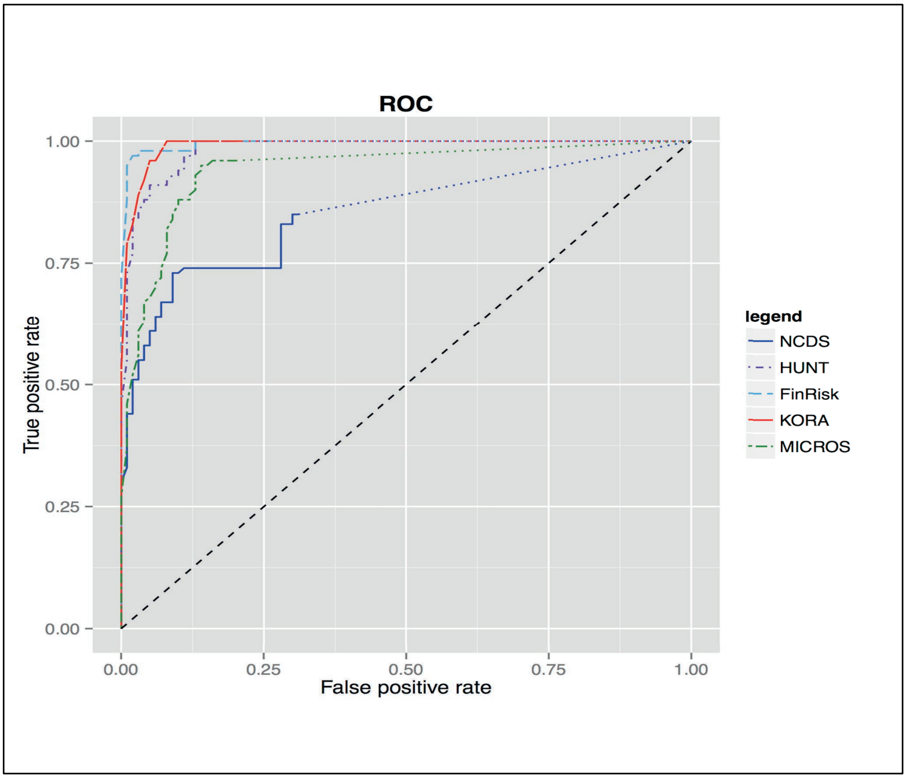


Figure 5 | **ROC curve.** Matching performance for 32 data elements in 5 different biobanks. Note that BiobankConnect only retrieves a subset of data elements based on the semantic/lexical similarity queries, therefore the ROC curves ends before reaching 1.00,1.00. For the remaining data elements we simulated a line of non-discrimination, indicated by dotted lines.

Rank order of final matches compared with expert decisions

We also evaluated BiobankConnect prioritization performance by evaluating the ranks of the best matches from the BioSHaRE project, i.e. what is the position of the match that the human experts chose from the longer lists of relevant matches. The median rank was 1 and the mean rank was 1.85. **Table 2** summarizes the frequencies of the ‘best matches’ per rank. The complete list of BioSHaRE best matches and BiobankConnect’s suggested matches is given in **Supp Table S1**.

Table 2 | **Ranking performance.** $P_{1,2}$ shows the rank of 191 expert selected 'best' matches within the automatic produced lists of relevant matches, using ontology annotations of the desired data elements or Lucene matching only, respectively. BiobankConnect predicted 'best' matches as first choice (rank 1) in 63.9% of the cases and within 'top 10' in 98.4% of the cases.

Rank	P_1 (using ontology)	***Cumulative*** P_1	P_2 (Lucene)	***Cumulative*** P_2
1	63.9% (n=122)	63.9% (n=122)	51.3% (n=98)	51.3% (n=98)
2	14.1% (n=27)	78.0% (n=149)	12.0% (n=23)	63.4% (n=121)
3	8.40% (n=16)	86.4% (n=165)	8.37% (n=16)	71.7% (n=137)
4	3.10% (n=6)	89.5% (n=171)	4.18% (n=8)	75.9% (n=145)
5	3.70% (n=7)	93.2% (n=178)	5.23% (n=10)	81.2% (n=155)
6	3.10% (n=6)	96.3% (n=184)	1.04% (n=2)	82.2% (n=157)
7	0.00% (n=6)	96.3% (n=184)	0.00% (n=0)	82.2% (n=157)
8	1.50% (n=3)	97.8% (n=187)	1.04% (n=2)	83.2% (n=159)
9	0.60% (n=1)	98.4% (n=188)	2.09% (n=4)	85.3% (n=163)
10	0.00% (n=0)	98.4% (n=188)	0.52% (n=1)	85.6% (n = 164)
≥10	0.00% (n=0)	98.4% (n=188)	3.66% (n=7)	89.5% (n=171)
Not found		1.60% (n=3)		10.5% (n=20)
Total		100% (n=191)		100% (n=191)

Contribution of ontology annotations

We compared the ranking of 'best' matches using ontological and Lucene lexical matching with using lexical matching only (see **Table 2)**. Out of 191 matches, using ontology annotations led to 17 matches that would otherwise have been missed, 28 large improvements (4.17 ranks on average) and 7 small decreases (1.71 ranks on average), which were significant changes (p-value 0.03; Wilcoxon rank-sum test, see **Supp Tables S4** and **S5**). In particular, the 1st rank category increased by 12.6% while other ranks hardly changed (between -1.50% to +2.60%).

2.6 Discussion

While the time spent using BiobankConnect is easily calculated, it is difficult to quantify the time spent by human experts on performing the same task. Instead, we can approximate the gain by estimating how much BiobankConnect reduces the number of data elements that need manual evaluation by an expert. Obviously, in an ideal world the expert would look at each available data element and decide if it is a suitable match for each of the desired data elements. In the worst case, each expert would have to visit on average half of the total data elements before the 'best' match is found. This would be a lot of work so a more realistic comparison is to assume some smart searching strategies. We used the Lucence string matching to simulate a best case where the expert would use advanced lexical searches. **Table 3** shows the average ranks of best matches per biobank using BiobankConnect (1.8, missing=3), Lucence string matching (2.8, missing=20) only, and random searching (3730) respectively. This suggests that BiobankConnect reduces the number of data elements that need to be evaluated by a factor of 1.5 to 2,000. The string-matching algorithms miss relevant elements due to non-standard descriptions or unexpected data elements that turn out to be valid proxies.

Table 3 | **BiobankConnect reduces the amount of data elements that need to be checked.** $R_{1,2,3}$ shows the average rank of the 'best' match when searching using BiobankConnect, using Lucence string matching only and random iteration, respectively.

Biobank *(number of elements)*	***R_1*** (via BiobankConnect)	***R_2*** (string-matching)	***R_3*** (random search)
Kora-gen (75)	1.5	1.8	36
MICROS (119)	2.0	1.3	59
FinRisk (223)	1.5	1.9	111
Hunt (353)	2.5	4.1	174
NCDS (516)	1.2	1.8	260
Prevend (6174)	2.2	4.3	3109
Average	1.8	2.7	3730
Missed elements	3	20	0

We wish to improve BiobankConnect and therefore investigated why recall was worse in, for example, the NCDS biobank and why some best matches were not ranked as top candidates. We discovered that bad matches were often caused by 'too many matches', 'repeated measurements', 'too specific questions', or 'complex proxy variable' (see **Supp Table S6**). We discuss these issues and suggest some solutions below.

- The issue of 'too many relevant matches' resulted in relatively low recall in NCDS. Scrutiny revealed this was caused by a large number of relevant matches for one particular data element. While for most desired data elements only 1-5 NCDS data elements were marked as relevant, 58 elements were relevant for 'EDU_HIGHEST' because they all cover some aspect of education. However, BiobankConnect only retrieved 11 out of 58, having a large impact on the calculation of recall.
- The issue with 'repeated measurements' occurred in Prevend, where data elements were measured multiple times at different time points. For example, for 'Current quantity of cigarettes smoked', there were

two data elements that had been manually matched: ‘V29_4’ with the description ‘Numbers of cigarettes per day’ and ‘V28_1’ with the description ‘Cigarettes or fine-cut tobacco in history or present’. V29_4 was ranked 2nd in the suggested list, whereas V28_1 was ranked 8th because there were another 6 data elements that had a similar description to V29_4. This search could be improved using ontology annotations that pinpoint the desired time points.

- The issue with ‘too specific data element’ occurred when matching ‘Current quantity of spirits/liquor consumed’ in MICROS. For example, descriptions of the manually determined matches were ‘Quantity of schnapps’ and ‘Previous quantity of schnapps’, in which ‘schnapps’ is an example of spirits/liquor. However, schnapps had not been defined in any of the ontologies on BioPortal, so it was not recognized as a special type of liquor and was therefore not mapped. This could be addressed by improving details in the current ontologies.
- The issue with ‘complex proxy variable’ was due to the proxy data elements used in matching being very difficult to find automatically. For example, ‘Fasting status’ and ‘Blood glucose level’ were measured separately in Prevend and, in addition, ‘Fasting status’ was derived from another two data elements: ‘When was the last meal?’ and ‘When was the last drink?’. Similarly, in NCDS, the data element ‘Blood glucose’ was not measured, but a human expert picked a proxy data element ‘Glycated hemoglobin’, which is known to correlate with plasma glucose. Matching for these data elements could be improved by using a new ontology that defines such complex relationships between biobank data elements.

In the current version of BiobankConnect, data elements are matched based only on the label or short description of the element, which may result in erroneous matching of some elements. However, biobanks contain more information that is not yet being used. For example, the data element ‘Blood pressure’ was recorded in all our biobanks, but the protocols used to measure blood pressure may differ across biobanks. If detailed protocol descriptions could be provided by the biobanks and incorporated into our system, the matches produced by BiobankConnect could be made more accurate. For the categorical data, information on the various categories could also be used to improve the match. Access to individual-level data could also employ statistical characteristics of the data to evaluate the pooling potential by comparing instance-based matching to schema matching. In addition, the use or development of more biobank-oriented ontologies might improve our system's performance. For example, the problem of 'too many matches' for education data elements could be alleviated by using a more specific ontology for the education parameters captured in biobanks.

Finally, we would like to be able to keep track of users' choices because this human expertise could provide important information to train our system and reproduce the findings thus far. For example, where 'Fasting glucose' was manually matched with a proxy variable 'Glycated hemoglobin' in NCDS, this relationship could be added to suitable ontologies, so that the information can be re-used for query and thereby developing BiobankConnect into a community knowledge base.

2.7 Conclusion

Within a matter of minutes BiobankConnect is able to find relevant data element matches with 0.75 precision at rank 1 and 0.74 recall at rank 10. The best matches are in the top-10 in 98.4% of the cases. BiobankConnect is therefore a useful tool to speed up the harmonization and integration of data across biobanks, with potential for use in other biomedical integration challenges. A demonstration and the open source software are available at http://www.biobankconnect.org.

Acknowledgments

We thank Dany Dorion for providing us with the data dictionaries of KORA, FINRISK, MICROS and NCDS biobanks and Kirsti Kvaløy for the data dictionary of HUNT biobank. We thank Jackie Senior for editing the final text.

Funding Statement

This work was supported by European Union Seventh Framework Programme (FP7/2007-2013) grant number 261433 (Biobank Standardisation and Harmonisation for Research Excellence in the European Union - BioSHaRE-EU) and grant number 284209 (BioMedBridges), TI Food and Nutrition grant number TIFN GH001, and BBMRI-NL grant number 184.021.007, a research infrastructure financed by the Netherlands Organization for Scientific Research (NWO).

Competing Interests Statement

The authors have no competing interests to declare.

Contributorship Statement

CP, MS, HH conceived the methods and designed the software. CP, MD, DH, KJV and MS implemented and tested the software. CP, HH, MS drafted the manuscript. All authors read and agreed with the software and the manuscript.

Chapter 3

SORTA: a System for Ontology-based Re-coding and Technical Annotation of biomedical phenotype data

Chao Pang,[1,3] Annet Sollie,[2] Anna Sijtsma,[3,4] Dennis Hendriksen,[1] Bart Charbon,[1] Mark de Haan,[1] Tommy de Boer,[1] Fleur Kelpin,[1] Jonathan Jetten,[1] K. Joeri van der Velde,[1] Nynke Smidt,[3,4] Rolf Sijmons,[2] Hans Hillege,[3] Morris A. Swertz[1,2,4]

1. Genomics Coordination Centre, University of Groningen, University Medical Centre Groningen; 2. University of Groningen, University Medical Centre Groningen, Department of Genetics, University of Groningen; 3. University Medical Centre Groningen, Department of Epidemiology, Groningen, The Netherlands; 4. LifeLines Cohort Study and Biobank, Groningen, The Netherlands

Database *2015, bav089. (September 2015)*

DOI: 10.1093/database/bav089

Abstract

There is an urgent need to standardize the semantics of biomedical data values, such as phenotypes, to enable comparative and integrative analyses. However, it is unlikely that all studies will use the same data collection protocols. As a result, retrospective standardization is often required, which involves matching of original (unstructured or locally coded) data to widely used coding or ontology systems such as SNOMED CT (clinical terms), ICD-10 (International Classification of Disease), and HPO (Human Phenotype Ontology). This data curation process is usually a time-consuming process performed by a human expert.

To help mechanize this process, we have developed SORTA, a computer-aided system for rapidly encoding free text or locally coded values to a formal coding system or ontology. SORTA matches original data values (uploaded in semicolon delimited format) to a target coding system (uploaded in Excel spreadsheet, OWL ontology web language or OBO open biomedical ontologies format). It then semi-automatically shortlists candidate codes for each data value using Lucene and n-gram based matching algorithms, and can also learn from matches chosen by human experts.

We evaluated SORTA's applicability in two use cases. For the LifeLines biobank, we used SORTA to recode 90,000 free text values (including 5,211 unique values) about physical exercise to MET (Metabolic Equivalent of Task) codes. For the CINEAS clinical symptom coding system, we used SORTA to map to HPO, enriching HPO when necessary (315 terms matched so far). Out of the shortlists at rank 1, we found a precision/recall of 0.97/0.98 in LifeLines and of 0.58/0.45 in CINEAS. More importantly, users found the tool both a major time saver and a quality improvement because SORTA reduced the chances of human mistakes. Thus, SORTA can dramatically ease data (re)coding tasks and we believe it will prove useful for many more projects.

Database URL: http://molgenis.org/sorta or as an open source download from http://www.molgenis.org/wiki/SORTA.

3.1 Introduction

Biobank and translational research can benefit from the massive amounts of phenotype data now being collected by hospitals and via questionnaires. However, heterogeneity between data sets remains a barrier to integrated analysis. For the BioSHaRE[49] biobank data integration project, we previously developed BiobankConnect[50], a tool to overcome heterogeneity in data *structure* by mapping data *elements* from the source database onto a target scheme. Here, we address the need to overcome heterogeneity of data *contents* by coding and/or recoding data *values*, i.e. mapping free text descriptions or locally coded data values onto a widely used coding system. In this 'knowledge-based data access', data is collected and stored according to local requirements while information extracted from the data is revealed using standard representations, such as ontologies, to provide a unified view[51].

The (re)coding process is essential for the performance of three different kinds of functions:

I) **Search and query.** The data collected in a research and/or clinical setting can be described in numerous ways with the same concept often associated with multiple synonyms, making it difficult to query distributed database systems in a federated fashion. For example, using standard terminologies, the occurrence of 'cancer' written in different languages can be easily mapped between databases if they have been annotated with same ontology term.

II) **Reasoning with data**. Ontologies are the formal representation of knowledge and all of the concepts in an ontology have been related to each other using different relationships, e.g. 'A is a *subclass of* B'. Based on these relationships, the computer can be programmed to reason and infer the knowledge[52]. For example, when querying cancer patients' records from hospitals, those annotated with 'Melanoma' will be retrieved because 'Melanoma' is specifically defined as a descendant of 'Cancer' in the ontology.

III) **Exchange or pooling of data across systems**. Ontologies can also be used to describe the information model, such as the MGED

(Microarray Gene Expression Data) ontology describing microarray experiments or hospital information coded using the ICD-10 (International Classification of Diseases) coding system, so that the data can easily flow across systems that use the same model[52].

The data (re)coding task is essentially a matching problem between a list of free text data values to a coding system, or from one coding system to another. Unfortunately, as far as we know, there are only a few software tools available that can assist in this (re)coding process. Researchers still mostly have to evaluate and recode each data value by hand, matching values to concepts from the terminology to find the most suitable candidates. Not surprisingly, this is a time-consuming and error-prone task. Based on our previous success in BioSHaRE, we were inspired to approach this problem using ontology matching and lexical matching[50]. We evaluated how these techniques can aid and speed-up the (re)coding process in the context of phenotypic data. In particular, we used our newly developed system, SORTA, to recode 5,210 unique entries for ‘physical exercise’ in the LifeLines biobank[5] and 315 unique entries for ‘physical symptoms’ (including terms that are similar, but not the same) in the Dutch CINEAS (www.cineas.org)[53] and HPO (Human Phenotype Ontology) coding systems for metabolic diseases.

Requirements

Several iterations of SORTA-user interviews resulted in the identification of the following user requirements:

1) Comparable similarity scores, e.g. scores expressed as a percentage, so users can easily assess how close a suggested match is to their data, and decide on a cut-off to automatically accept matches.
2) Support import of commonly used ontology formats (OWL/OBO) for specialists and Excel spread sheets for less technical users.
3) Fast matching algorithm to accommodate large input datasets and coding systems.

4) Online availability so users can recode/code data directly and share with colleagues without need to download/install the tool.

5) Maximize the sensitivity to find candidate matches and let users decide on which one of them is the ‘best’ match.

6) Enable complex matching in which not only a text string is provided but also associated data elements such as labels, synonyms and annotations, e.g. [label: Hearing impairment, synonyms:(Deafness, Hearing defect)].

Approaches

Two types of matching approaches have been reported in the literature: lexical matching and semantic matching. **Lexical matching** is a process that measures the similarity between two strings[12]. Edit-distance[54], n-gram[55] and Levenshtein distance[56] are examples of string-based algorithms that focus on string constituents and are often useful for short strings, but they do not scale up for matching large numbers of entity pairs. Token-based techniques focus on word constituents by treating each string as a bag of words. An example of these techniques is the vector space model algorithm[57], in which each word is represented as a dimension in space and a cosine function is used to calculate the similarity between two string vectors. Lexical matching is usually implemented in combination with a normalization procedure such as lowering case, removing stop words (e.g. ‘and’, ‘or’, ‘the’) and defining word stems (e.g. ‘smoking’ → ‘smoke’). **Semantic matching** techniques search for correspondences based not only on the textual information associated to a concept (e.g. description) but also on the associative relationships between concepts (e.g. subclass, ‘is-a’)[12]. In these techniques, for example, ‘melanoma’ is a good partial match for the concept called ‘cancer’. Because our goal is to find the most likely concepts matching data values based on their similarity in description, lexical-based approaches seem most suitable.

One of the challenges in the (re)coding task is the vast number of data values that need to be compared, which means that the matcher has to find correspondences between the Cartesian product of the original data values and the codes in the desired coding system. High-throughput algorithms are needed to address this challenge and two methods have been developed to deal with the matching problem on a large scale. The Early Pruning Matching Technique[58] reduces search space by omitting irrelevant concepts from the matching process, e.g. the ontology concept (label:hearing impairment, synonyms[deafness, hearing defect, congenital hearing loss]) that does not contain any words from the search query ‘protruding eye ball’ are eliminated. The Parallel Matching Technique[58] divides the whole matching task into small jobs and the matcher then runs them in parallel, e.g. 100 data values are divided into 10 partitions that are matched in parallel with ontologies.

Existing tools

We found several existing tools that offered partial solutions, see **Table 1**. Mathur and Joshi [59] described an ontology matcher, Shiva, that incorporates four string-matching algorithms (Levenshtein distance, Q-grams, Smith Waterman and Jaccard), any of which could be selected by users for particular matching tasks. They used general resources like WordNet and Online Dictionary to expand the semantics of the entities being matched. Cruz [60] described a matcher, Agreement Maker, in which lexical and semantic matchers were applied to ontologies in a sequential order and the results were combined to obtain the final matches. At the lexical matching stage, Cruz [60] applied several different kinds of matchers, string-based matches (e.g. edit distance and Jar-Winkler) and an internally revised token-based matcher, then combined the similarity metrics from these multiple matchers. Moreover the philosophy behind this tool is that users can help make better matches in a semi-automatic fashion that are not possible in automatic matching [60]. Jiménez-Ruiz and Cuenca Grau [61] described an approach where: I) they used lexical matching to compute an initial set of matches; II) based on these initial matches, they took advantage of semantic reasoning methods to discover more matches in the class hierarchy, and III) they used indexing technology to increase the efficiency of computing the match correspondences between ontologies. Peregrine [62] is an indexing engine or tagger that recognizes concepts within human readable text, and if terms match multiple concepts it tries to disambiguate BioPortal[42], the leading search portal for ontologies, provides the BioPortal Annotator that allows users to annotate a list of terms with pre-selected ontologies. While it was useful for our use cases, it was limited because it only retrieves perfect matches and terms with slightly different spellings cannot be easily matched (e.g. ‘hearing impaired’ vs. ‘hearing impairment’)[63]. In addition, BioPortal Annotator’s 500-word limit reduces its practical use when annotating thousands of data values. Finally, ZOOMA[22] enables semi-automatic annotation of biological data with selected ontologies and was closest to our needs. ZOOMA classifies matches as ‘Automatic’ or ‘Curation required’ based on whether or not there is manually curated knowledge that supports the suggested matches. ZOOMA does not meet our requirements in that it does not provide similarity scores for the matches, does not prioritize recall over precision (i.e. ZOOMA matches are too strict for our needs), and does not handle partial/complex matches. For example, in ZOOMA, the OMIM (Online Mendelian Inheritance in Man) term ‘Angular Cheilitis’ could not be partially matched to the HPO term ‘Cheilitis’ and ‘Extra-Adrenal Pheochromocytoma’ could not be matched to the HPO term ‘Extraadrenal pheochromocytoma’ because of the hyphen character.

Table 1 | **Comparison of existing tools with SORTA.** ZOOMA and BioPortal Annotator were the closest to our needs.

	SORTA	BioPortal annotator	ZOOMA	Shiva	Agreement maker	LogMap	Peregrine
Comparable similarity score	Y	N	N	N	Y	Y	N
Import code system in ontology format	Y	Y	Y	Y	Y	Y	Y
Import code system in excel format	Y	N	N	N	N	N	N
Uses lexical index to improve performance	Y	Y	Y	N	N	Y	Y
Code/Recode data directly in the tool	Y	N	N	N	Y	N	N
Tool available as online service	Y	Y	Y	N/A	N/A	N/A	N
Support partial matches	Y	N	N	Y	Y	Y	N
Match complex data values	Y	N	N	Y	Y	Y	N
Learns from curated dataset	Y	N	Y	N	N	N	N

Y represents Yes; N represents No; N/A represents unknown

3.2 Method

Based on our evaluation of existing tools, we decided to combine a token-based algorithm, Lucene[45], with an n-gram-based algorithm. Lucene is a high-performance search engine that works similarly to the Early Pruning Matching Technique. Lucene only retrieves concepts relevant to the query, which greatly improves the speed of matching. This enables us to only recall suitable codes for each value and sort them based on their match. However, the Lucene matching scores are not comparable across different queries making it unsuitable for human evaluation. Therefore, we added an n-gram-based algorithm as a second matcher, which allows us to standardize the similarity scores as percentages (0-100%) to help users understand the quality of the match and to enable a uniform cut-off value.

We implemented the following three steps. First, coding systems or ontologies are uploaded and indexed in Lucene to enable fast searches (once for each ontology). Second, users create their own coding/recoding project by uploading a list of data values. What users get back is a shortlist of matching concepts for each value that has been retrieved from the selected coding system based on their lexical relevance. In addition, the concepts retrieved are matched with the same data values using the second matcher, the n-gram-based algorithm, to normalize the similarity scores to values from 0-100%. Finally, users apply a %-similarity-cut-off to automatically accept matches and/or manually curates the remaining codes that are assigned to the source values. Finally, users download the result for use in their own research. An overview of the strategy is shown in **Figure 1**. We provide a detailed summary below. Users upload coding sources such as ontologies or terminology lists to establish the knowledge base. Ontologies are the most frequently used source for matching data values, but some of the standard terminology systems are not yet available in ontology formats. Therefore, we allow users to not only upload ontologies in OWL and OBO formats, but also import a ‘raw knowledge base’ stored in a simple Excel format which includes system ID, concept ID, and label (see **Table 2)**. The uploaded data is then indexed and stored locally to enable rapid matching.

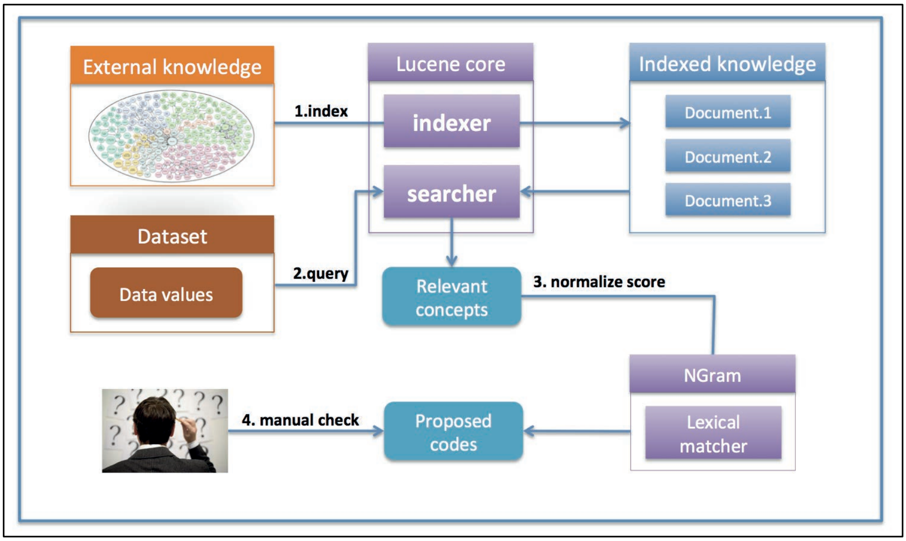


Figure 1 | **SORTA overview.** The desired coding system or ontology can be uploaded in OWL/OBO and Excel and indexed for fast matching searches. Data values can be uploaded and then automatically matched with the indexed ontology using Lucene. A list of the most relevant concepts is retrieved from the index and matching percentages are calculated using the n-gram algorithm so that users can easily evaluate the matching score. Users can choose the mappings from the suggested list.

Table 2 | **Example of how to upload a coding system and a coding/recoding target.** This example shows an Excel file with MET (Metabolic Equivalent of Task), a system developed to standardize physical activity, in which each concept ID includes a list of different sports representing specific amounts of energy consumption.

Concept ID	***Concept Label***	***System ID***
02060	cardio training	MET
02020	bodypump	MET
18310	swimming	MET
15430	kung fu	MET
15350	hockey	MET
12150	running	MET

To match data values efficiently, we used the Lucene search index with the default snowball stemmer and a standard filter for stemming and removing stop words. A code/ontology concept is evaluated as being a relevant match for the data value when it or its corresponding synonyms (if available) contain at least one word from the data value. The assumption in this strategy is that the more words a concept's label or synonyms contain, the more relevant Lucene will rank it, and therefore the top concepts on the list are most likely to be the correct match. However, the snowball stemmer could not stem some of the English words properly, e.g. the stemmed results for 'placenta' and 'placental' were 'placenta' and 'placent', respectively. To solve this problem, we enabled fuzzy matching with 80% similarity and this allowed us to maximize the number of relevant concepts retrieved by Lucene.

Lucene also provides matching scores that are calculated using a cosine similarity between two weighted vectors [64], which takes the information content of words into account, e.g. rarer words are weighted more than common ones. However, after our first user evaluations we decided not to show Lucene scores to users for two reasons. First, Lucene calculates similarity scores for any indexed document as long as it contains at least one word from the query. Documents that have more words that match the query, or contain words that are relatively rare, will get a higher score. Secondly, the matching results produced by different queries are not comparable because the scales are different [65] making it impossible to determine the 'best' cut-off value above which the suggested matches can be assumed to be correct.

We therefore decided to provide an additional similarity score that ranges from 0-100% by using an n-gram calculation between the data value and the relevant concepts retrieved by Lucene. In this n-gram-based algorithm, the similarity score is calculated for two strings each time. The input string is lowercased and split by whitespace to create a list of words, which are then stemmed by the default snowball stemmer. For each of the stemmed words, it is appended with '^' at the beginning and '$' at the end, from which the bigram tokens are generated, e.g. ^smoke$ → [^s, sm, mo, ok, ke, e$]. All the bigram tokens are pushed to a list for the corresponding input string with duplicated tokens allowed. The idea is that the more similar two strings are, the more bigram tokens they can share. The similarity score is the product of number of shared bigram tokens divided by the sum of total number of bigram tokens of two input strings as follows,

$$Similarity = \frac{Number\ of\ shared\ bigram\ tokens \times 2}{Number\ of\ bigram\ tokens_{S1} + \ Number\ of\ bigram\ tokens_{S2}}$$

Because we were only interested in the constituents of the strings being compared, the order of the words in strings does not change the score. We also considered only using the n-gram calculation, but that would require calculation of all possible pairwise comparisons between all data values and codes, which would greatly slow down the process.

Ultimately both algorithms were combined because Lucene is very efficient in retrieving relevant matches while our users preferred n-gram scores because they are easier to compare. Combining Lucene with the n-gram-based algorithm is an optimal solution in which the advantages of both methods complement each other while efficiency, accuracy and comparability of scores are preserved.

To code the data values, the data can be uploaded as a simple comma separate value file or copy/pasted into the text area directly in SORTA. The uploaded data is usually a list of simple string values, however in some cases it also can be complex data values containing information other than a simple label. For these cases, SORTA allows inclusion of descriptive information such as synonyms and external database identifiers to improve the quality of the matched results shown in **Table 3**.

Table 3 | **Example of how to upload data values and coding/recoding source)**. At minimum, one column of values should be provided: the first column with the header ‘Name’. Additional optional columns that start with ‘Synonym_’ can contain the synonyms for input values. Other optional column headers can contain other identifiers, e.g. in this example OMIM.

Name *(required)*	***Synonym_1****(optional)*	***OMIM*** *(optional)*
2,4-dienoyl-CoA reductase deficiency	DER deficiency	222745
3-methylcrotonyl-CoA carboxylase deficiency	3MCC	210200
Acid sphingomyelinase deficiency	ASM	607608

For each of the data values, a suggested list of matching concepts is retrieved and sorted based on similarity. Users can then check the list from the top downwards and decide which of the concepts should be selected as the final match. However, if the first concept on the list is associated with a high similarity score, users can also choose not to look at the list because they can confidently assume that a good match has been found for that data value. By default, 90% similarity is the cut-off above which the first concept on the retrieved list is automatically picked as the match for the data value and stored in the system. Below 90% similarity, users are required to manually check the list to choose the final match. The cut-off value can be changed according to the needs of the project, e.g. a low cut-off of 70% can be used if the data value was collected using free text because typos are inevitably introduced during data collection.

3.3 Results

We evaluated SORTA in various projects. Here we report two representative matching scenarios where the original data values were either free text (case 1) or already coded, but using a local coding system (case 2). In addition, as a benchmark, we generated matches between HPO, NCIT (National Cancer Institute Thesaurus), OMIM (Online Mendelian Inheritance in Man) and DO (Disease Ontology) and compared the matches with existing cross references between these two (case 3)

Case 1: Coding unstructured data in the LifeLines biobank

Background

LifeLines is a large biobank and cohort study started by the University Medical Centre Groningen, the Netherlands. Since 2006, it has recruited 167,729 participants from the northern region of the Netherlands[5]. LifeLines is involved in the EU BioSHaRE consortium and one of the joint data analyses being conducted by BioSHaRE is the 'Healthy Obese Project' (HOP) that examines why some obviously obese individuals are still metabolically healthy[6]. One of the variables needed for the HOP analysis is physical activity but, unfortunately, this information was collected using a Dutch questionnaire containing free text fields for types of sports. Researchers thus needed to match these to an existing coding system: the Ainsworth compendium of physical activities[66]. In this compendium each code matches a metabolic equivalent task (MET) intensity level corresponding to the energy cost of that physical activity and defined as the ratio of the metabolic rate for performing that activity to the resting metabolic rate. One MET is equal to the metabolic rate when a person is quietly sitting and can be equivalently expressed as:

$$1\,MET \equiv 1\,\frac{kcal}{kg\times h} \equiv 4.184\,\frac{kJ}{kg\times h}$$

A list of 800 codes has been created to represent all kinds of daily activities with their corresponding energy consumption[66]. Code 1015, for example, represents ‘general bicycling’ with a MET value of 7.5. The process of matching the physical activities of LifeLines data with codes is referred to as coding.

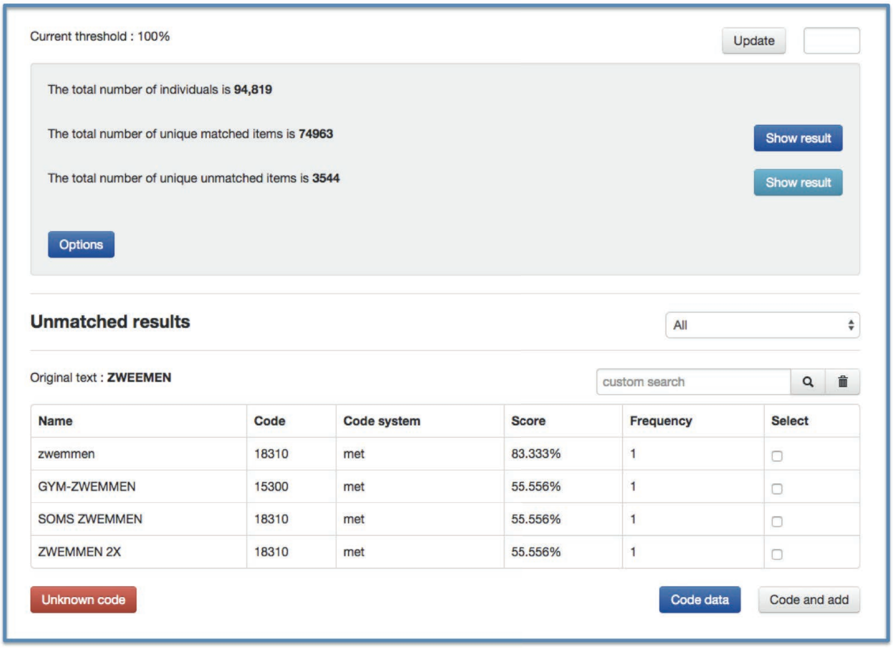
Challenges and motivation

There were two challenges in this task. First, the physical activities were collected in Dutch and therefore only researchers with a good level of Dutch could perform the coding task. Second, there were data for more than 90,000 participants and each participant could report up to four data values related to ‘Sport’ that could be used to calculate the MET value. In total, there were 80,708 terms (including 5,211 unique terms) that needed to be coded. We consulted with the researchers and learned that they typically coded data by hand in an Excel sheet or by syntax in SPSS, and for each entry they needed to cross-check the coding table and look up the proper code. While this approach is feasible on a small scale (<10,000 participants), it became clear it would be too much work to manually code such a massive amount of data. Hence, we used our SORTA coding system.

To train SORTA, we reused a list of human-curated matches between physical activities described in Dutch and the codes that were created for a previous project. We used this as the basis to semi-automatically match the new data from LifeLines. An example of the curated matches is shown in **Table 2** and the complete list can be found at **Supplementary material: Lifelines_MET_mappings.xlsx**. Moreover, we have enhanced SORTA with an upload function to support multiple ‘Sport’-related columns in one harmonization project. This can be done as long as the column headers comply with the standard naming scheme, where the first column header is

'Identifier' and other column headers start with string 'Sport_', e.g. 'Sport_1' and 'Sport_2'.

Figure 2 shows an example of manually coding the physical activity 'ZWEMMEN' (Swimming) with MET codes, in which a shortlist of candidates were retrieved by SORTA and the first item of the list selected as the true match. Each time the manual curation process produced a new match, this new knowledge could be added to the knowledge base to be applied to all future data values. This is an optional action because data values (especially those filled in by participants of the study) sometimes contain spelling errors that should not be added to the knowledge base.



Current threshold : 100%

The total number of individuals is **94,819**

The total number of unique matched items is **74963**

The total number of unique unmatched items is **3544**

Unmatched results

Original text : **ZWEEMEN**

Name	Code	Code system	Score	Frequency	Select
zwemmen	18310	met	83.333%	1	□
GYM-ZWEMMEN	15300	met	55.556%	1	□
SOMS ZWEMMEN	18310	met	55.556%	1	□
ZWEMMEN 2X	18310	met	55.556%	1	□

Figure 2 | **Example of coding a physical activity.** A list of MET codes was matched with input and sorted based on similarity scores, from which the proper code can be selected to recode the input. If none of the candidate codes is suitable, users can either search for codes manually or decide to use 'Unknown code'. If the button 'Code data' is clicked, the input is recoded only with the selected code. If the button 'Code and add' is clicked, the input is recoded and the input gets added to the code as a new synonym. The example is a typo of the Dutch word for "swimming". zwemmen = swimming, zwemmen 2x = twice a week, soms zwemmen = occasional swimming, gym-zwemmen = water gym.

Evaluation

With the assistance of SORTA, all of the data values have been coded by the researcher who is responsible for releasing data about physical activity in the

LifeLines project. The coding result containing a list of matches was used as the gold standard for the following analysis, in which we evaluated two main questions: I) How far could the previous coding round improve the new matching results? II) What is the best cut-off value above which the codes selected by SORTA can be confidently assumed to be correct matches to a value?

SORTA's goal is to shortlist good codes for the data values so we first evaluated the rank of the correct manual matches because the higher they rank, the less manual work the users need to perform. Our user evaluations suggested that as long as the correct matches were captured in the top 10 codes, the researchers considered the tool useful. Otherwise, based on their experience, users changed the query in the tool to update the matching results.

Re-use of manually curated data from the previous coding round resulted in an improvement in SORTA's performance with recall/precision at rank 1st increasing from 0.59/0.65 to 0.97/0.98 and at rank 10th from 0.79/0.14 to 0.98/0.11 (see **Figure 3** and **Table 4**). At the end of the coding task, about 97% of correct matches were captured at rank 1st with users only needing to look at the first candidate match.

We included use of an n-gram-based algorithm to provide users with an easily understood metric with which to judge the relevance of the proposed codes on a scale of 1-100%, based on the n-gram match between value and code (or a synonym thereof). **Supplementary Table S1** suggests that, in the LifeLines case, 82% similarity is a good cut-off for automatically accepting the recommended code because 100% of the matches produced by the system were judged by the human curator to be correct matches. Because LifeLines data is constantly being updated (with new participants, and with new questionnaire data from existing participants every 18 months), it would be really helpful to recalibrate the cut-off value when the tool is applied anew.

Table 4 | **Precision and recall for the LifeLines case study.** In total, 90,000 free text values (of which 5,211 were unique) were recoded to physical exercise using MET coding system. The table shows recall and precision per position in the SORTA result before coding (using only the MET score descriptions) and after coding (when a human curator had already processed a large set of SORTA recommendations by hand).

	Before coding			After coding		
Rank	R	P	F	R	P	F
1	0.59	0.65	0.62	0.97	0.98	0.97
2	0.66	0.39	0.49	0.97	0.50	0.66
3	0.71	0.29	0.41	0.97	0.34	0.50
4	0.74	0.24	0.36	0.97	0.26	0.41
5	0.76	0.21	0.33	0.97	0.21	0.35
6	0.77	0.19	0.30	0.97	0.18	0.30
7	0.78	0.17	0.28	0.97	0.15	0.26
8	0.78	0.16	0.27	0.98	0.14	0.25
9	0.78	0.14	0.24	0.98	0.12	0.21
10	0.79	0.14	0.24	0.98	0.11	0.20
11	0.79	0.13	0.22	0.98	0.10	0.18
12	0.79	0.12	0.21	0.98	0.09	0.16
13	0.79	0.12	0.21	0.98	0.09	0.16
14	0.79	0.12	0.21	0.98	0.08	0.15
15	0.79	0.11	0.19	0.98	0.08	0.15
16	0.79	0.11	0.19	0.98	0.07	0.13
17	0.79	0.11	0.19	0.98	0.07	0.13
18	0.80	0.11	0.19	0.98	0.06	0.11
19	0.80	0.10	0.18	0.98	0.06	0.11
20	0.80	0.10	0.18	0.98	0.06	0.11
30	0.80	0.10	0.18	0.98	0.04	0.08
50	0.80	0.09	0.16	0.98	0.03	0.06

R recall; P precision; F F-measure;

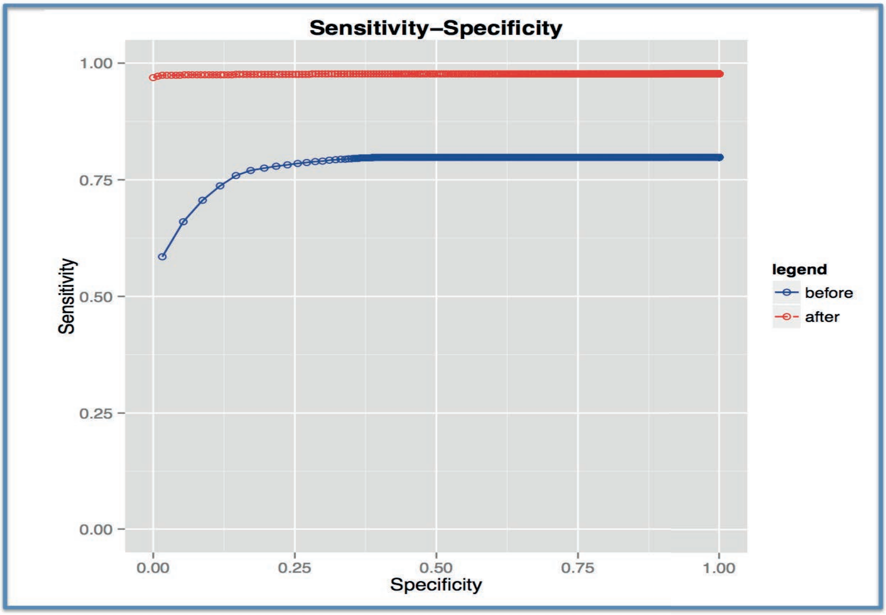


Figure 3 | **Receiver operating characteristic (ROC) curves evaluating performance on LifeLines data.** Blue represents the performance before the researcher recoded all the LifeLines data. During coding, the researcher introduced new knowledge to the database and if a similar dataset was uploaded again (e.g. second rounds of the same questionnaire), the coding performance greatly improved as shown by the red curve.

Case 2: Recoding from CINEAS coding system to HPO ontology

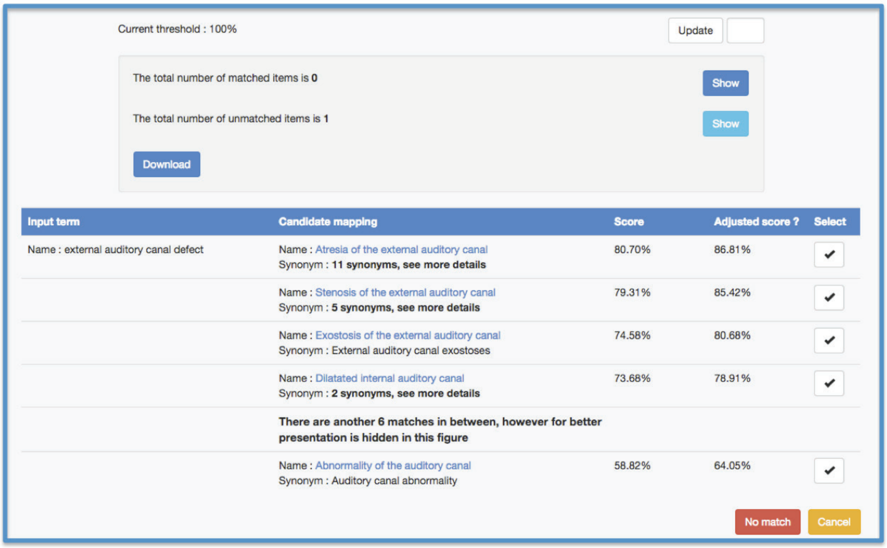
Background

CINEAS is the Dutch centre for disease code development and its distribution to the clinical genetics community (www.cineas.org)[53]. This centre was initiated by the eight clinical genetics centres responsible for genetic counselling and diagnostics in the Netherlands in 1992[67]. CINEAS codes are used in daily practice by Dutch clinical geneticists and genetic counsellors to assign diseases and clinical symptoms to patients. The 63rd edition of CINEAS now lists more than 5,600 diseases and more than 2,800 clinical symptoms. The challenge was to match and integrate (or recode) the CINEAS clinical symptom list with HPO in order to use one enriched standardized coding system for future coding of patients' symptoms and to obtain interoperability for CINEAS codes already registered in local systems all over the country. The metabolic diseases obtained from CINEAS disease list, which has become an independent project called The Dutch Diagnosis

Registration Metabolic Diseases (DDRMD, https://ddrmd.nl/)[67], will be matched with Orphanet ontology in the future.

Challenge and motivation

The previous strategy of CINEAS curators was to search HPO via BioPortal, however, tracking possible candidate terms meant making written notes or keeping a digital registry on the side, tracking methods that are time-consuming, prone to human errors and demand a lot of switching between tools or screens. Therefore, SORTA was brought into the project. **Figure 4** shows an example of a data value 'external auditory canal defect' and a list of HPO ontology terms as candidate matches. While none of them is a perfect match for the input term, the top three candidates are the closest matches, but are too specific for the input. Scrutiny by experts revealed that 'Abnormality of auditory canal' could be a good 'partial' match because of its generality.



Current threshold : 100%

The total number of matched items is **0**

The total number of unmatched items is **1**

Input term	Candidate mapping	Score	Adjusted score ?	Select
Name : external auditory canal defect	Name : Atresia of the external auditory canal Synonym : **11 synonyms, see more details**	80.70%	86.81%	✓
	Name : Stenosis of the external auditory canal Synonym : **5 synonyms, see more details**	79.31%	85.42%	✓
	Name : Exostosis of the external auditory canal Synonym : External auditory canal exostoses	74.58%	80.68%	✓
	Name : Dilatated internal auditory canal Synonym : **2 synonyms, see more details**	73.68%	78.91%	✓
	There are another 6 matches in between, however for better presentation is hidden in this figure			
	Name : Abnormality of the auditory canal Synonym : Auditory canal abnormality	58.82%	64.05%	✓

Figure 4 | **Example of matching the input value 'external auditory canal defect' with HPO ontology terms**. A list of candidate HPO ontology terms was retrieved from the index and sorted based on similarity scores. Users can select a mapping by clicking the 'v' button. If none of the candidate mappings are suitable, users can choose the 'No match' option.

Evaluation

In an evaluation study, the first 315 clinical symptoms out of 2,800 were re-coded by a human expert, in which 246 were matched with HPO terms while 69 could not be matched. In addition, we performed the same matching task using BioPortal Annotator and ZOOMA because these existing tools seemed most promising (see **Table 5)**. We further investigated which cut-off value can be confidently used to assume that the automatic matches are correct by calculating precision and recall for all possible n-gram cut-offs (0-100%). **Supplementary Table S2** shows 89% to be a good cut-off value for future CINEAS matching tasks because above this value all of the suggested matches are correct with 100% precision.

Case 3: Benchmark against existing matches between ontologies

We downloaded 700 existing matches between HPO and DO concepts, 1148 matches between HPO and NCIT concepts, and 3631 matches between HPO and OMIM concepts from BioPortal. We used the matching terms from DO, NCIT and OMIM as the input values and HPO as the target coding system and generated matches using SORTA, BioPortal Annotator and ZOOMA. **Supplementary Table S3** shows that all three tools managed to reproduce most of the existing ontology matches with SORTA slightly outperforming the other two by retrieving all of the ontology matches. Scrutiny revealed that SORTA was able to find the complex matches, where data values and ontology terms consist of multiple words, and some of which are concatenated, e.g. matching ‘propionic acidemia’ from DO with ‘Propionicacidemia’ from HPO. We also noticed that beyond the 1st rank, precision in SORTA is lower than the other two (with the highest precision in ZOOMA). In addition, we investigated what proportion of data values could be automatically matched at different cut-offs. **Supplementary Table S4** shows that at similarity score cut-off of 90%, SORTA recalled at least 99.6% of the existing matches with 100% precision across all three matching experiments.

Table 5 | **Comparison of SORTA, BioPortal and ZOOMA.** Evaluation based on the CINEAS case study in which 315 clinical symptoms were matched to Human Phenotype Ontology. The table shows the recall/precision per position in SORTA, BioPortal Annotator and ZOOMA. N.B. both BioPortal Annotator and ZOOMA have a limitation that they can only find exact matches and return a maximum of three candidates.

	SORTA			BioPortal			ZOOMA		
Rank	R	P	F	R	P	F	R	P	F
1	0.58	0.45	0.51	0.34	0.54	0.42	0.17	0.63	0.27
2	0.69	0.27	0.39	0.35	0.44	0.39	0.17	0.60	0.26
3	0.73	0.19	0.30	0.35	0.44	0.39	0.18	0.60	0.28
4	0.76	0.15	0.25	N/A	N/A	N/A	N/A	N/A	N/A
5	0.78	0.13	0.22	N/A	N/A	N/A	N/A	N/A	N/A
6	0.81	0.11	0.19	N/A	N/A	N/A	N/A	N/A	N/A
7	0.81	0.09	0.16	N/A	N/A	N/A	N/A	N/A	N/A
8	0.83	0.08	0.15	N/A	N/A	N/A	N/A	N/A	N/A
9	0.83	0.08	0.15	N/A	N/A	N/A	N/A	N/A	N/A
10	0.85	0.07	0.13	N/A	N/A	N/A	N/A	N/A	N/A
11	0.85	0.06	0.11	N/A	N/A	N/A	N/A	N/A	N/A
12	0.85	0.06	0.11	N/A	N/A	N/A	N/A	N/A	N/A
13	0.86	0.06	0.11	N/A	N/A	N/A	N/A	N/A	N/A
14	0.86	0.05	0.09	N/A	N/A	N/A	N/A	N/A	N/A
15	0.87	0.05	0.09	N/A	N/A	N/A	N/A	N/A	N/A
16	0.87	0.05	0.09	N/A	N/A	N/A	N/A	N/A	N/A
17	0.87	0.05	0.09	N/A	N/A	N/A	N/A	N/A	N/A
18	0.88	0.04	0.08	N/A	N/A	N/A	N/A	N/A	N/A
19	0.88	0.04	0.08	N/A	N/A	N/A	N/A	N/A	N/A
20	0.88	0.04	0.08	N/A	N/A	N/A	N/A	N/A	N/A
30	0.89	0.03	0.06	N/A	N/A	N/A	N/A	N/A	N/A
50	0.92	0.02	0.04	N/A	N/A	N/A	N/A	N/A	N/A

N/A not applicable; R recall; P precision; F F-measure;

3.4 Discussion

In RESULTS section, we have evaluated SORTA in three different use cases. It has shown that SORTA could indeed help human experts in performing the (re)coding tasks in terms of improving the efficiency and user evaluations of SORTA were very positive, but there was much debate among co-authors on the combination of Lucene-based matching with n-gram post-processing. As mentioned in the Method section, Lucene scores were not really informative for users, but the order in which the matching results were sorted by Lucene seemed better thanks to the cosine similarity function that takes information content into account. After applying the n-gram-based algorithm, this order was sometimes changed. To evaluate this issue we performed the same matching tasks using Lucene and Lucene + n-gram. In the case of coding LifeLines data, the performances were quite similar and the inclusion of n-gram did not change the order of the matching results, see **Supplementary material: PrecisionRecallLifeLines.xlsx**. However, in the case of matching HPO terms, there was a large difference in precision and recall as shown in **Figure 5** and **Supplementary material PrecisionRecallCINEAS.xlsx**. Lucene alone outperformed the combination of the two algorithms. We hypothesize that this may be caused by Lucene's use of word inverse document frequency (IDF) metrics, which are calculated for each term (t) using the following formula:

$$idf(t) = 1 + \log\left(\frac{totalNumber_{docs}}{docFreq + 1}\right)$$

where docFreq is the number of documents that contain the term.

We checked the IDFs for all the words from input values for the HPO use case and **Supplementary Figure S5** shows the large difference in the information carried by each word. This suggested that, to improve the usability of the tool, we should allow users to choose which algorithm they wish to use to sort the matching results, an option that we will add in the near future. We also explored if we could simply add information content to the n-gram scoring mechanism to make the ranks consistent by redistributing the contribution of each of the query words in the n-gram score based on the IDF. For example, using n-gram the contribution of the word ‘joint’ in the query string ‘hyperextensibility hand joint’ is about 18.5% because ‘joint’ is 5/27 letters. However, if this word is semantically more important, results matching this word should have a higher score. We therefore adapted the n-gram algorithm to calculate the IDF for each of the words separately, calculate the average, and reallocate the scores to the more important words as follows:

$$Score_{reallocate} = \frac{length_{common\_word}}{length_{all\ words}} \times \frac{IDF_{average} - IDF_{common\_word}}{IDF_{average}}$$

$$Score_{common\_word} = \frac{length_{common\_word}}{length_{all\ words}} - Score_{reallocate}$$

$$Score_{important\_word} = \frac{length_{important\_word}}{length_{all\ words}} + \sum Score_{reallocate} \times \frac{IDF_{important\_word}}{\sum IDF_{important\_words}}$$

Common_word is defined as having an IDF that is lower than IDF$_{average}$
Important_words is defined as the IDF that is higher than IDF$_{average}$

This resulted in an improvement of recall compared to naive n-gram scoring at rank 10th from 0.79 to 0.84 (for details see **Supplementary material: comparision_ngram_lucene.xlsx**), and the summarized comparison is provided via receiver operating characteristic (ROC) curve in **Figure 5**.

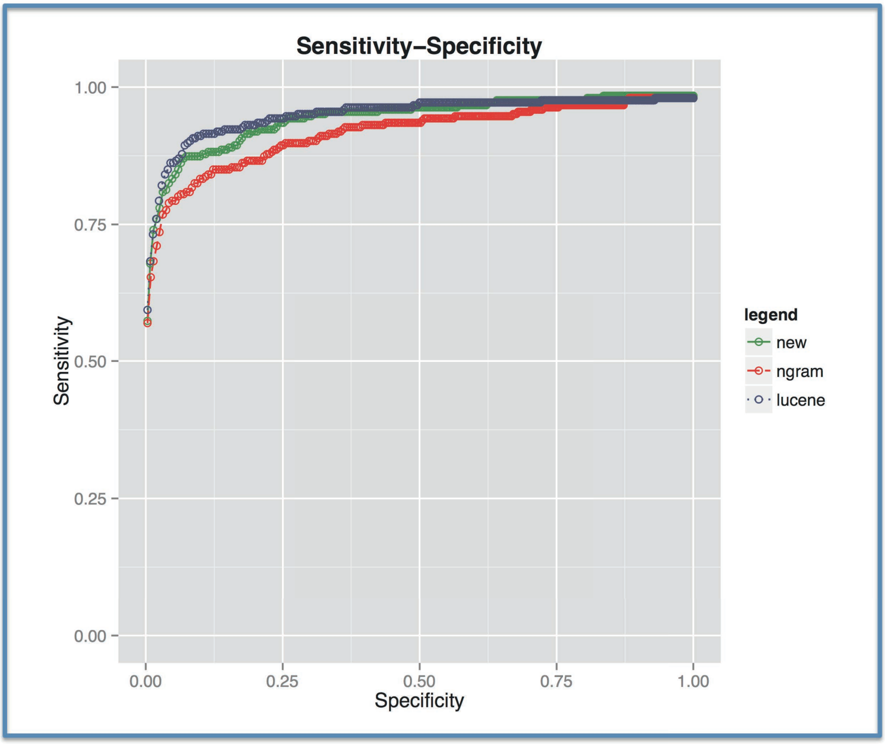


Figure 5 | **Performance comparison for matching HPO terms among three algorithms**. Lucene (blue line), combination of Lucene + n-gram (red) and combination of Lucene + n-Gram + inverse document frequency (green).

However, Lucene still outperforms this metric and we speculate that this can be explained by the fundamental difference between the underlying scoring functions. The n-gram score is more sensitive to the length of input strings than Lucene and it is quite possible that two strings do not share any of the words but share similar bigram tokens, especially when dealing with long strings. Consequently, the n-gram-based algorithm might find more false positives than Lucene. However, in practice, the number of data values to be coded/recoded is quite large and the benefit of using an n-gram score cut-off value above which all the suggested matches are automatically selected outweighs this drawback.

Another issue was whether we could make better use of all the knowledge captured in ontologies. We noticed in some matching examples that related terms that come from the same ontological cluster tend to show up together in the matching results. For example, Figure 4 shows that the input term 'external auditory canal defect' is not matched to any of the top three candidates because they are too specific and hence we have to take the more general ontology term 'Auditory canal abnormality', which is actually ranked 11th, as the match even though this term is in fact the parent of the three top candidates. This indicates that if the input value is not matched by any of the candidates with a high similarity score and the candidates contain clusters of ontology terms, the parent ontology term should probably be selected as the best match (which is similar to the way human curators make decisions on such matches). However, translating this knowledge into an automatic adaptation of matching a score is non-trivial and something we plan to work on in the future.

3.5 Conclusions

We developed SORTA as a software system to ease data cleaning and coding/recoding by automatically shortlisting standard codes for each value using lexical and ontological matching. User and performance evaluations demonstrated that SORTA provided significant speed and quality improvements compared to the earlier protocols used by biomedical researchers to harmonize their data for pooling. With increasing use, we plan to dynamically update the precision and recall metrics based on all users' previous selections so that users can start the matching tasks with confident cut-off values. In addition, we plan to include additional resources such as WordNet for query expansion to increase the chance of finding correct matches from ontologies or coding systems. Finally, we also want to publish mappings as linked data, for example as nanopublications [68] (http://nanopub.org), so they can be easily reused. SORTA is available as a service running at http://molgenis.org/sorta. Documentation and source code can be downloaded from http://www.molgenis.org/wiki/SORTA under open source LGPLv3 license.

Acknowledgements

This work was supported by the European Union Seventh Framework Programme (FP7/2007-2013) grant number 261433 (Biobank Standardisation and Harmonisation for Research Excellence in the European Union - BioSHaRE-EU) and grant number 284209 (BioMedBridges). It was also supported by BBMRI-NL, a research infrastructure financed by the Netherlands Organization for Scientific Research (NWO), grant number 184.021.007. We thank Anthony Brookes of Leicester University who contributed the abbreviation ‘SORTA’, and Kate Mc Intyre and Jackie Senior for editing the manuscript.

Chapter 4

MOLGENIS/connect: a system for semi-automatic integration of heterogeneous phenotype data with applications in biobanks

Chao Pang[1,2], David van Enckevort[1], Mark de Haan[1], Fleur Kelpin[1], Jonathan Jetten[1], Dennis Hendriksen[1], Tommy de Boer[1], Bart Charbon[1], Erwin Winder[1], Joeri K. van der Velde[1], Dany Doiron[3], Isabel Fortier[3], Hans Hillege[2], Morris A. Swertz[1,2,*]

[1]University of Groningen, University Medical Center Groningen, Genomics Coordination Center, Department of Genetics, Groningen, the Netherlands. [2]University of Groningen, University Medical Center Groningen, Department of Epidemiology, Groningen, the Netherlands. [3]Research Institute of the McGill University Health Centre and Department of Medicine, McGill University, Montreal, Canada.

Bioinformatics *32, 2176-2183 (March 2016)*

DOI: 10.1093/bioinformatics/btw155

Abstract

Motivation: While the size and number of biobanks, patient registries and other data collections are increasing, biomedical researchers still often need to pool data for statistical power, a task that requires time-intensive retrospective integration.

Results: To address this challenge, we developed MOLGENIS/connect, a semi-automatic system to find, match and pool data from different sources. The system shortlists relevant source attributes from thousands of candidates using ontology-based query expansion to overcome variations in terminology. Then it generates algorithms that transform source attributes to a common target DataSchema. These include unit conversion, categorical value matching and complex conversion patterns (e.g. calculation of BMI). In comparison to human-experts, MOLGENIS/connect was able to auto-generate 27% of the algorithms perfectly, with an additional 46% needing only minor editing, representing a reduction in the human effort and expertise needed to pool data.

Availability: Source code, binaries and documentation are available as open-source under LGPLv3 from http://github.com/molgenis/molgenis and www.molgenis.org/connect.

Contact: m.a.swertz@rug.nl

4.1 Introduction

Biobanks, patient registries and other human data collections have become an indispensable resource to better understand the epidemiology and biological mechanisms of disease. While these collections have grown to include data from over 100,000s of individuals, many research questions still require data from multiple collections to reach sufficient statistical power or to achieve sufficient numbers of subjects having rare (disease) characteristics. To make data integration easy, all collections would ideally use the same data collection protocols and questionnaires. In practice however, biobanks collect different data because of differences in their scientific goals. For integration to be valid, data must be compared and harmonized before combined analyses are carried out (Fortier et al., 2011).

Substantial efforts are now underway to make data 'inferentially equivalent' or 'harmonized' as a basis for pooled analysis. The Maelstrom Research group has taken the lead in defining protocols for retrospective data integration (https://www.maelstrom-research.org/)[11]. Within the BioSHaRE project, we have re-used and refined this protocol to harmonize and integrate 90 variables from 9 biobanks as a basis for pooled analysis [20]. This research-question-driven approach consists of three steps:

1. **Defining the target DataSchema**: the list of targeted variables necessary to address the research questions in a specific study;
2. **Matching biobank schemas to the target DataSchema**: match data elements from participating data sources/biobanks to the variables in the target DataSchema;
3. **Generating of Extract-Transform-Load algorithms**: define the algorithms that take the matched source data elements as the input and convert these data values to the target DataSchema for data integration.

Existing biomedical data integration tools still require significant manual effort and technical skill. For example, Maelstrom uses Opal software for biobank pooling with a professional team to find mappings and create algorithms, available at http://www.obiba.org/pages/products/opal/ [19]. Similarly, clinical/translational data warehouses tranSMART and i2b2 require knowledgeable analysts to manually identify mappings, based on which ETL developers implement the programmatic transformations [69,70]. To alleviate this burden, we previously presented BiobankConnect, a system to semi-automatically match data elements from biobanks to target variables (Pang, Hendriksen, et al., 2015). In this paper, we introduce an additional system to semi-automatically define the transformation algorithms to produce an integrated dataset. We have wrapped all functions described above into an integrated user interface, MOLGENIS/connect, to support research teams through the entire integration procedure.

4.2 Methods

We have used the Maelstrom Research harmonization protocol as the basis for our system. **Figure 1** provides an overview of its main components.

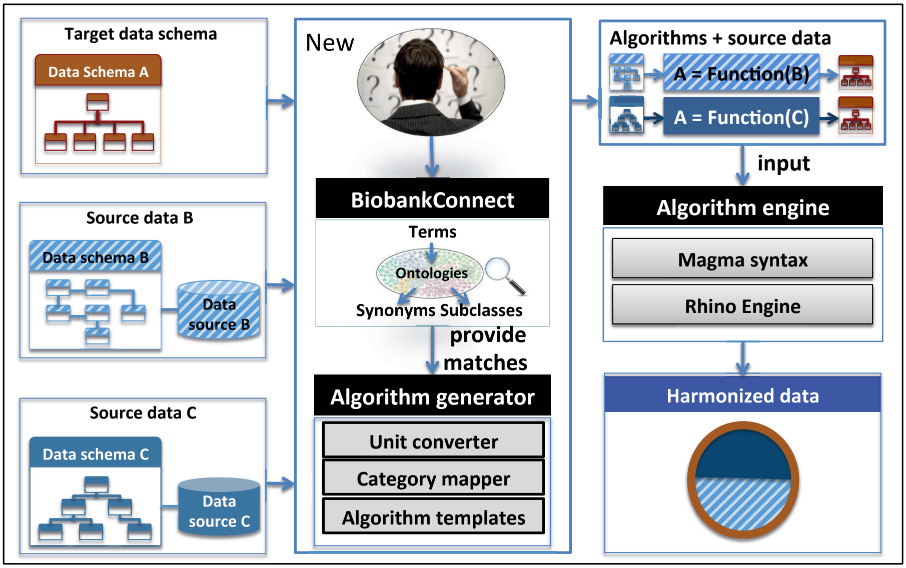


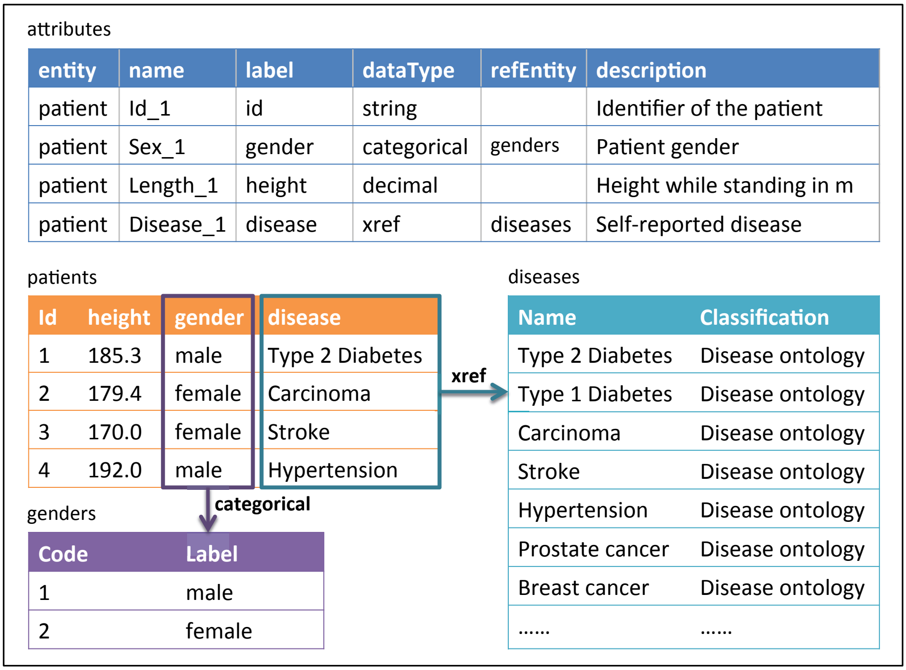
Figure 1 | **The overview of the framework of MOLGENIS/connect**.

First, we implemented a metadata model component that allows users to upload, view and visualize the target DataSchemas as well as the data of the source biobanks. Second, we incorporated a semantic search facility to shortlist candidate source data element matches to each variable in the target

DataSchema. Third, an integration algorithm generator incorporates algorithm templates, semantic searches, category convertors and a unit convertor.

Metadata model

To load both the target DataSchema as well as the various biobank data models (i.e. data dictionaries), we have designed a flexible meta-model called Entity Model Extensible (EMX), the documentation is available at http://molgenis.github.io/documentation/ [71]. This model evolved from Observ-OM, which has been proven to model all kinds of biomedical data [41]. EMX is a lightweight version of Observ-OM in which only two types of information (Entity and Attribute) are needed to sufficiently describe a dataset. Attributes are features that can be observed such as 'disease', 'gender' and 'height', and which are often referred to as 'metadata' by researchers. In EMX, an attribute ideally contains the following information: a unique name, a pre-defined data type (e.g. string, integer, decimal), a human readable label, a detailed description of the attribute and how it can be used, and categories or cross-references (xrefs) if the data type is categorical or a relationship (e.g. 'Gender attribute' has two categories, 'Male' and 'Female'). Entities are definitions of tables that define groups of attributes as columns and data (entity instances) as rows. The relations of entities and attributes are described in **Figure 2**. In the rest of this paper, we will refer to both of the variables of the target DataSchema and the data elements of the source (biobank) as 'attributes'.



attributes

entity	name	label	dataType	refEntity	description
patient	Id_1	id	string		Identifier of the patient
patient	Sex_1	gender	categorical	genders	Patient gender
patient	Length_1	height	decimal		Height while standing in m
patient	Disease_1	disease	xref	diseases	Self-reported disease

patients

Id	height	gender	disease
1	185.3	male	Type 2 Diabetes
2	179.4	female	Carcinoma
3	170.0	female	Stroke
4	192.0	male	Hypertension

genders

Code	Label
1	male
2	female

diseases

Name	Classification
Type 2 Diabetes	Disease ontology
Type 1 Diabetes	Disease ontology
Carcinoma	Disease ontology
Stroke	Disease ontology
Hypertension	Disease ontology
Prostate cancer	Disease ontology
Breast cancer	Disease ontology
……	……

Figure 2 | **Example of the EMX data upload format.** Data can be uploaded using Excel Metadata describing the columns of each data sheet (i.e. ‘entity’) that must be provided in a special ‘attributes’ sheet. Data values are stored in ordinary sheets (e.g. ‘patients’). The ‘categorical’ gender attribute and the ‘xref’ disease attribute refer to another two sheets, ‘genders’ and ‘diseases’ (omitted for readability).

Semi-automatic source-to-target attribute matching

Standard practice for identifying candidate biobank attributes for pooled analyses has been to manually go through all data attributes of all biobanks, an extremely time-consuming process. To automate this step, we used our previously published BiobankConnect method [50]. It combines the Information Retrieval System of Lucene, available at https://lucene.apache.org/core/ [45], with query expansion to automatically shortlist good candidate attributes. It consists of I) query expansion [72] in which attributes of the target DataSchema are (semi-) automatically annotated (‘expanded’) with ontology terms, whose synonyms and subclasses are collected to create a list of semantically identical or similar terms that get added to the original query to find other relevant source attributes and II) retrieving relevant attributes in which the ‘expanded’ target attributes are matched against the biobank attributes using Lucene, and matched candidates are sorted based on Lucene scores for human experts to choose from, as described in [50].

Transformation syntax

To create an executable data integration procedure, the rules for transforming data from source to target attributes need to be encoded in a computer algorithm. These algorithms transform attribute values from the source datasets to the statistically equivalent attribute value required in the target DataSchema. The simplest algorithm simply renames the source attribute, e.g. transforming ‘length’ (in LifeLines) to ‘height’ in the target DataSchema. More advanced algorithms can implement unit conversions, recode categories or execute more advanced formulas like a body mass index (BMI) calculation.

For the implementation of the transformation algorithms, we have used the ‘Magma’ syntax [18], available at http://wiki.obiba.org/display/OPALDOC/Magma+Javascript+API, which is a domain-specific programming language for data harmonization that was used in BioSHaRE. Magma is a JavaScript library that works similar to jQuery, a popular JavaScript framework. To access values, the name of attributes can be wrapped in brackets and a dollar sign, e.g. $(‘var’). There are many methods available in Magma which can be called by chaining calls to the attribute accessor, e.g. $(‘var’).div(2). We have implemented the most commonly used methods including div(), times(), plus(), map(), pow(), unit() and toUnit(). In addition we have created an algorithm generator, which consists of a unit conversion algorithm generator, a categorical values algorithm generator and a complete algorithm generator, described below.

Unit conversion algorithm generator

One of the recurring challenges in data harmonization is harmonizing units. Detecting units in attribute metadata can be difficult because different forms of units are used to describe the same parameter in different databases, e.g. ‘meter’ is used to describe the attribute ‘Height in meter’ in one database while ‘cm’ is used in describing the attribute ‘Body length in cm’ in another. Because no suitable algorithm generator could be found, we have developed a new two-step method for unit convertor generation. First, unit terms that occur in the label of target attributes and/or source attributes are annotated with the Units of Measurement Ontology (UO). Labels of attributes and target attributes are tokenized by whitespace and matched against terms in the UO using Lucene (analogous to how BiobankConnect does attribute matching). To prevent false positives, we accept only exact matches for unit detection. Second, we have used the unit converter software library developed by JScience [73], which is implemented based on The Unified Code for Units of Measure http://www.unitsofmeasure.org/trac [74], for international standard units and commonly used non-standard units, available at http://jscience.org/. This has a list of conversion rules for units that are compatible, e.g. cm = m × 100 or g = kg × 1000. For example, to convert units from ‘centimeter’ to ‘meter’ for the attribute ‘Height’, the terms ‘centimeter’ and ‘meter’ are automatically annotated with ontology terms UO:centimeter and UO:meter, respectively, based on the formal name and synonyms of the units. The formal symbols of these two units (cm and m) collected from the UO are then parsed to JScience, in which the suitable rule is found for converting ‘cm’ to ‘m’ and incorporated into the algorithm template. We implemented two different syntaxes for unit conversions: using a chain of explicit methods, e.g., **$(‘Height’).unit(‘cm’).toUnit(‘m’).value()**, or more by generating the necessary calculation formula, e.g., **$(‘Height’).div(100).value()**. In the case of composite units or derived units such as kg/m2, we first break them into the smallest units (atomic units), then compare the atomic units with units of matched attributes individually, and finally convert the units accordingly. For example, the target attribute BMI (kg/m2) is matched to source attributes **height in cm** and **weight in gram**. The term kg/m2 is broken apart into a set of atomic units, kg and m, which become the standard units because they are detected/derived from the target attribute, the cm and gram units detected from source attributes are then converted accordingly.

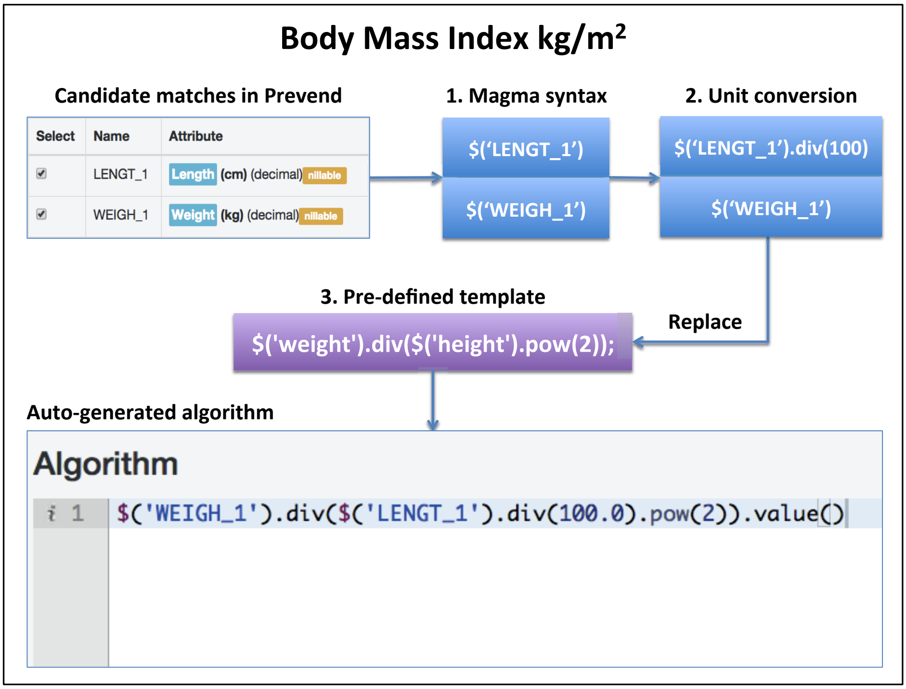
Categorical values matching generator

Another recurring challenge is to generate algorithms that convert between categorical values. For this, we explored matching categories automatically and identified three different types of categories that need to be matched:

- <u>Matching categories using lexical similarity</u>: To find lexically similar categories, we calculate the pairwise n-gram similarity scores between all target and source categories. For each source category, the target category that yielded the best n-gram similarity score is automatically selected as the best match. For example, the target attribute (Gender:‘0=Male,1=Female’) and the source attribute (SEX:‘1=Male,2=Female’) have the same category labels but different category codes, the system matches two sets of category labels onto each other based on the n-gram-based string matching algorithm and with the final result $(‘Gender’)=$(‘SEX’).map({‘1’:’0’, ’2’:’1’}). Thus source category 1 and 2 are matched to target category 0 and 1, respectively.
- <u>Matching categories that represent frequencies:</u> After scrutinizing many biobank attributes and the target attributes, we realized that there are a class of attributes that describes the frequencies of certain activities or food consumption. **Supplementary Table S6** shows an example of matching attributes for consumption of potatoes. The categories contain two types of information, time units and frequencies, which can be extracted using regular expressions, e.g. 2-4 times a week has an average frequency 3 (2-4) and the time unit week. The first step is to convert both the target and source categories to quantifiable amounts; the second step is to find the closest target amount category for each source amount category. Because categories are often not matched one-to-one, the algorithm is allowed to have multiple source amounts matched to one target amount. The matching category function is implemented in Java using JScience library [73].
- <u>Matching categories based on pre-defined rules:</u> In **Supplementary Table S7**, we show a list of custom rules for matching categories that we have hard-coded into the system.

Overall algorithm generator

The creation of algorithms is a tricky task and nearly impossible for those inexperienced in programming. Therefore, as a last step, we created a generator that assembles the complete algorithms. Moreover, we have provided a catalogue of templates for more complex algorithms such as ‘BMI calculation’, which can be found in the Supplementary material **javascript_magma.xls**. Each template defines its source and target attributes. These matching templates will be proposed to the user if one or more of the matched attributes relates to this template, e.g. ‘height’ or ‘weight’ in the case of BMI.



Select	Name	Attribute
☑	LENGT_1	Length (cm) (decimal) nillable
☑	WEIGH_1	Weight (kg) (decimal) nillable

```
$('WEIGH_1').div($('LENGT_1').div(100.0).pow(2)).value()
```

Figure 3 | **Example of algorithm generation for target attribute BMI from the Prevend data source** (1) a transformation template is generated from the candidate matches (using Magma syntax), (2) the template is automatically edited based on unit conversion rules if applicable, (3) the software evaluates if more complex algorithm templates can be used. Based on two good candidate matches and the desired ‘BMI’ target, a previously used BMI conversion algorithm is proposed that incorporates the unit conversion rules (e.g. from ‘cm’ to ‘m’ because BMI is recorded as composite unit kg/m^2).

Figure 3 summarizes the process of generating the complete algorithm using the example of the target attribute ‘Body Mass Index’ from source biobank Prevend. It consists of the following steps: I) the system looks in its database to find the available algorithm template for BMI, II) it uses the BiobankConnect algorithm to generate a list of relevant attributes, III) it applies the unit conversion algorithms towards kg/m2 (e.g. LENGT_1 was measured using centimeter (cm) rather than the standard unit meter (m) and therefore needs to be converted), and IV) the building blocks within the BMI template are replaced with the matched attributes using the string-matching algorithm (n-gram)(e.g. ‘weight’ was matched with ‘WEIGHT_1:Weight (kg)’ and ‘height’ was matched with LENGT_1: Length (cm) based on the best lexical similarity scores).

4.3 Implementation

We have implemented above methods into a seamless user workflow: (1) users upload a target DataSchema and the source biobank data, (2) users then create a mapping project and select target DataSchema and data sources, (3) MOLGENIS/Connect automatically generates all matches and conversion algorithms for all data sources and all target attributes, (4) the user curates each of the matches and algorithms using the algorithm editor and preview tool and (5) MOLGENIS/Connect generates the integrated dataset. We describe each step in detail below. The integration tool has been built on top of the MOLGENIS software suite and reuses its basic functions (upload, metadata viewer, data explorer, permission system) [40]. MOLGENIS is a Java/Maven web application implemented using MySql and ElasticSearch as back-end and HTML5, Bootstrap, jQuery, ReactJS as front-end. The source code is available at https://github.com/molgenis.

Upload and view target DataSchema and data sources

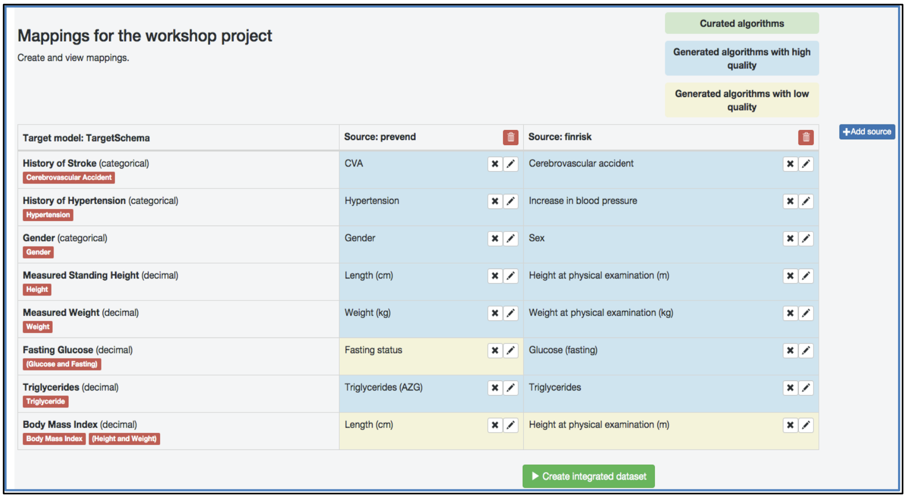
In this step, users upload target DataSchema and source data via the standard MOLGENIS upload. For this purpose, we use the 'EMX' format (Molgenis, 2014), a spreadsheet-based format to describe and upload tabular datasets and definition of their schemas that can be edited directly using Microsoft Excel or text editor (CSV files). For the target DataSchema, one spreadsheet is required that defines ‘attributes’ of the target DataSchema such as name, description and data type (see ‘attributes’ sheet in **Figure 2**). For each biobank, two spreadsheets are required: a ‘attributes’ metadata sheet just like the target DataSchema that defines the attributes of each dataset and one or more dataset sheets where each column matches the attributes and each row is, e.g., data on each biobank participant (see ‘your data' table in **Figure 2**). The data that has been uploaded can be viewed and filtered using MOLGENIS data explorer.

Create a mapping project

In this step users start a new mapping project with the desired DataSchema as the target and the biobank datasets as the sources. Once these are selected, the system will generate an overview of attribute matches (described below).

Generate overview of attribute mappings from source to target DataSchema

In this step the system generates a complete overview of all target attributes (shown in the first column) and all the matches from the source attributes (shown in the following columns), see **Figure 4.**



Mappings for the workshop project

Create and view mappings.

Curated algorithms
Generated algorithms with high quality
Generated algorithms with low quality
+Add source

Target model: TargetSchema	Source: prevend	Source: finrisk
History of Stroke (categorical) Cerebrovascular Accident	CVA	Cerebrovascular accident
History of Hypertension (categorical) Hypertension	Hypertension	Increase in blood pressure
Gender (categorical) Gender	Gender	Sex
Measured Standing Height (decimal) Height	Length (cm)	Height at physical examination (m)
Measured Weight (decimal) Weight	Weight (kg)	Weight at physical examination (kg)
Fasting Glucose (decimal) (Glucose and Fasting)	Fasting status	Glucose (fasting)
Triglycerides (decimal) Triglyceride	Triglycerides (AZG)	Triglycerides
Body Mass Index (decimal) Body Mass Index (Height and Weight)	Length (cm)	Height at physical examination (m)

Figure 4 | **Mapping project overview.** The attributes of the target DataSchema are shown on the left of the table. The columns contain matching attributes from each of the sources. New source data can be added by clicking the '+Add source' button. Attribute matches and conversion algorithms are automatically generated and colour coded: green indicates the algorithm has been curated by the user, blue indicates the algorithms are generated with high confidence (perfect match in semantic search) and yellow indicates the system predicts that auto-generated algorithms are of low quality (partial match in semantic search).

When a user selects a new data source, the system automatically generates candidate matches. Each match can be edited and tested using the algorithm editor described below. To open this view, users click on the pencil icon located in any of the cells. For this purpose, we have refactored the BiobankConnect system, which uses ontology terms to generate the candidate matches [50]. Based on user feedback, we learned that manual annotation of target attributes with ontologies previously required was too labour-intensive. We have, therefore, now included automatic annotation in which the label and description of the target attributes are used to find ontology terms in all available ontologies (e.g. NCI, SNOMED CT and MeSH) in the database.

Edit and test data transformations

In this step the user can edit the integration algorithm, see **Supplementary Figure S8**. This is the heart of the system and consists of three components: (1) the source attribute selector, (2) the algorithm editor and (3) the result preview.

In the source attribute selector (shown on the left of the screen) shortlists candidate attributes sorted by lexical matching scores between the ontology terms associated to the target attribute and label or description of the source attributes. The words from the ontology terms are highlighted in each attribute label or description. Based on the importance of the highlighted words, users can immediately determine whether the candidates generated are good matches for the target attribute or not. In the example in **Supplementary Figure S9a**, the words *blood* and *pressure* are highlighted in the attribute 'Mean blood pressure' and it is clear that this attribute is related but not the same as 'Hypertension'. If no good candidates are shown, the user can enter terms in the semantic search box to quickly find additional attributes using the syntax **term1 or term2** (e.g. weight or gender), see **Supplementary Figure S9b**. These query terms are matched with ontology terms to enable expanded query.

In the algorithm editor (shown in the middle), the user sees the auto-generated algorithm for the selected attribute (or multiple attributes) using the Magma/JavaScript syntax (see methods section). We mostly dealt with two types of target attributes: numeric attributes whose value can either be integer or decimal, e.g. the value for 'height' is a decimal number, and categorical attributes which only have a limited number of allowed values, e.g. values for 'gender' written in the JSON-like (http://www.json.org/) [75] format {code=0,label=male}, {code=1, label=female}. To generate algorithms for these target attributes, we usually need one source attribute, although sometimes the values of multiple attributes need to be combined, e.g. values for ‘BMI’ must be generated via ‘height’ and ‘weight’. Other data types supported include Date, Boolean, String and Text (see EMX documentation).

In the result preview (shown on the right of the screen), the user sees a subset of the results of the converted data and how many of the data conversions failed, e.g. because of syntax errors. This allows users to rapidly test and correct their conversion algorithms.

Create the derived dataset and explore the results

Having defined the algorithms in Magma/JavaScript as described above, users can execute the transformation process from within the mapping project overview. The data conversion engine is implemented using Rhino and the R interface with Rcurl and rjson, where Rcurl is used to retrieve data in JSON [75] format and convert it to a DataFrame object in R. A new dataset is then created that stores values in the target DataSchema. Users can access the data through MOLGENIS data explorer where advanced filtering function and visualization capability are offered. The integrated data can be downloaded in comma-separated values (CSV) and Microsoft Excel. We also provide the R Application Programming Interface (R-API), which allows users to access data in the R statistical environment (see MOLGENIS documentation), and HTTP REST/JSON interfaces to integrate with other software.

4.4 Results

We performed a qualitative evaluation by applying the software in active BioSHaRE, BBMRI and RD-Connect harmonization projects and a quantitative evaluation by comparing the auto-generated algorithms with the manually curated algorithms within the BioSHaRE Healthy Obese Project [6].

Matching numeric attributes

In the example shown in **Supplementary Figure S10a**, the target attribute ‘Measured Standing Height’ was matched to source attributes in the LifeLines biobank [5]. The first source attribute suggested, ‘Height in cm’, is used by default in generating the algorithm. The unit ‘cm’ was detected by the system in the source attribute whereas there was no mention of unit in the target attribute, therefore the target unit was assumed to be the same as the source attribute and unit conversion was not needed. Algorithms are executed automatically whenever users change the algorithm syntax in the editor; an updated preview of algorithm results is provided to evaluate.

Matching categorical attributes

Supplementary Figure S10b shows another example, in which the target attribute and the source attribute were both categorical. We implemented the Magma map({c1:c1’, c2:c2’….}) function to match categories of the target attribute and source attribute onto each other. A category-matching editor is demonstrated, where two sets of categories could be easily matched by selecting target categories from the dropdown menus. The results from the matching editor were converted to the Magma syntax so users could easily create matching functions without writing complex algorithms.

Evaluation of algorithm generator

We compared the output of the auto-generated transformation algorithms with manually curated algorithms for all 90 target attributes from the BioSHaRE Healthy Obese Project [6] and three of the biobanks (LifeLines, Prevend and Mitchelstown) for which we had the participant-level data values (184 algorithms in total). We evaluated the performance of semantic search and algorithm generation separately.

To evaluate the semantic search, we defined three result categories: perfect search, good search and bad search. A search result is ‘perfect’ when the human-matched source attribute was ranked 1st in the system-suggested list. A search result is ‘good’ when all human-matched source attributes can be found within top 20 of the suggested list. We chose this threshold because there were a few target attributes for which HOP research assistants used more than 10 source attributes. For example, there are 16 source attributes related to the target attribute ‘current consumption of meat product’ in Mitchelstown.

To evaluate the algorithm generator, we also defined three categories (perfect, good and bad). Algorithms were classified as ‘perfect’ when the auto-generated algorithms were the same as or functionally equivalent to manually created ones (i.e. when the algorithms yields the same target values when executed on the source data set). Algorithms were ‘good’ when they were almost correct but still required the users to fix them by hand. For example, when half of the categorical values were correctly matched between the source and the target attributes, but some additional matches also needed to be added by hand to complete the algorithm. An algorithm is evaluated to be ‘bad’ when the algorithm needs to be completely replaced by a human-edited version.

Table 1. Summary of the quality measures of algorithm generator and semantic search (in percentages)

	Perfect algorithms	Good algorithms	Bad algorithms	Total
Perfect search	51 (27.7%)	31 (16.8%)	3 (1.6%)	85 (46.1%)
Good search	18 (9.8%)	13 (7.1%)	17 (9.2%)	48 (26.1%)
Bad search	18 (9.8%)	12 (6.5%)	21 (11.4%)	51 (27.7%)
Total	87 (47.3%)	56 (30.4%)	41 (22.3%)	184 (100.0%)

Cells are color-coded to represent the amount of human input (manual work) required to fix the matching, with green being the easiest and red being the most difficult.

Table 1 summarizes the quantitative evaluation (the complete data can be found in the Supplementary material **Evaluation_results.xlsx)**: 27.7% of the algorithms generated were immediately equivalent to the manually created ones (perfect search, perfect algorithm); 9.8% of the algorithms generated where perfect, but only after users chose the right source attributes from the list of candidates (good search, perfect algorithm); 16.8% of the algorithms generated were partially correct and required users to modify them (perfect search, good algorithm); also we considered (good search, good algorithm), (bad search, perfect algorithm) and (perfect search, bad algorithm) to be useful. Thus, in total, 73% of the results were deemed useful (summing up the green color-coded cells in **Table 1,** 27.7+16.8+1.6+9.8+7.1+18=73).

4.5 Discussion & Future work

In the RESULTS section we demonstrated that MOLGENIS/connect can help users can quickly identify relevant source attributes and that the program auto-generates mostly useful data integration algorithms. Here we discuss potential areas of improvement.

Domain-specific improvements

Table 2. Quality measures of algorithm generator and semantic search in percentages, grouped by attribute topic

	Algorithm generator			**Semantic search**		
	Perfect	Good	Bad	Perfect	Good	Bad
Diet (10)	50%	40%	10%	70%	30%	0%
Disease (14)	86%	14%	0%	71%	29%	0%
Drink (8)	0%	38%	63%	50%	38%	13%
Education (17)	0%	82%	18%	65%	35%	0%
Food (42)	88%	5%	7%	14%	33%	52%
General (18)	28%	50%	22%	50%	11%	39%
Job (8)	0%	100%	0%	25%	0%	75%
Measurement (42)	62%	17%	21%	74%	10%	17%
Medication (11)	0%	36%	64%	27%	36%	36%
Smoking (14)	14%	21%	64%	14%	57%	29%
Total (184)	47%	30%	22%	46%	26%	28%

The numbers between brackets indicate the number of target attributes.

To obtain more insights into the cases for which the system performs well and the cases for which the system needs improvement, we have grouped all the target attributes into 10 areas of information: Diet, Disease, Alcohol use, Education, Food, Employment, physical and laboratory measurement, Medication, Tobacco use and General (e.g. Age, Gender). We summarize the performance of the algorithm generator as well as semantic search per topic in **Table 2** and **Figure 5**, for further details see **Supplementary Table S11.**

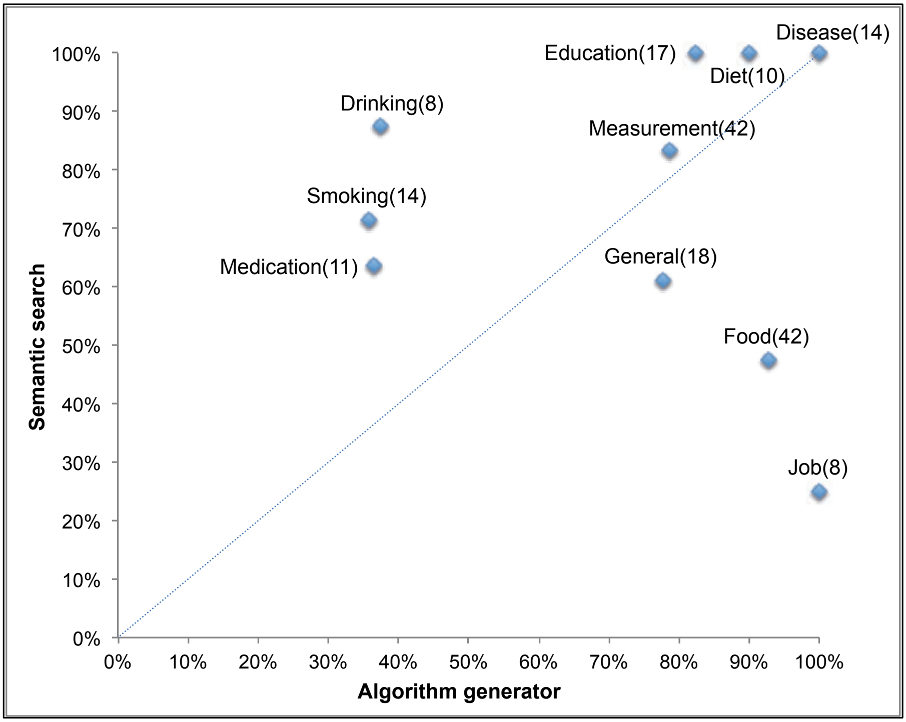


Figure 5 | **Scatter plot visualizing the success rates of algorithm generator and semantic search per attribute domain.** The X-axis and Y-axis represent 'useful algorithm' (defined as when the algorithms generated are correct or partially correct) and 'useful search' (defined as when the matched source attributes found fall within top 20 of the suggested list) categories of algorithm generator and semantic search in Table 2. The numbers in parenthesis are the number of attributes for the corresponding topics.

Figure 5 indicates that semantic search doesn't perform well on 'Food and 'Job' while algorithm generator needs improvement for 'Medication', 'Smoking' and 'Drinking'. Smoking and Drinking turned out to be very difficult to handle because how these attributes are defined in different biobanks varies in description and structure. There are more than 40 smoking-related attributes in LifeLines versus only 3 in Prevend. As a consequence, it was very difficult for semantic search to identify 'the one attribute' among many similar ones. Further, because there were few recurring patterns, the algorithm generator did not know how to generate the algorithms even though the source attributes were provided. We originally thought that the attribute Medication would be well standardized across biobanks due to the use of ATC code. In practice, some biobanks still use internally defined terminology to record medication information, making it more challenging to integrate medication data automatically. On the other hand, rather complex Food and Job target attributes scored unexpectedly 'good' in algorithm generation.

Semantic search is currently limited because we only used small subsets of SNOMED CT and NCI Thesaurus ontologies (for performance reasons). The search capability may be further improved by using the complete version of those ontologies. For instance, the target attribute 'Current Consumption Frequency of Poultry and Poultry Products' was matched to the source attribute Breaded chicken through manual matching, but semantic search missed this match due to the lack of knowledge of such terminology. The relation 'Chicken is_subclass_of Poultry' is stated explicitly in full SNOMED CT and search results could be greatly improved by incorporating such information. Other challenges in mapping attributes are the problem of family history, e.g. 'parental diabetes' which was discussed in [50], and of negation, e.g. 'I do not smoke' is considered relevant to the target attribute 'quantity of cigarette smoked'. One of the potential solutions would be to highlight the negative words in a specific colour in the suggested source attributes, such as not, never and don't, so users can immediately choose to skip those attributes.

Complex algorithms

Although semantic search and algorithm generator seem to work well, the algorithm template functionality is still limited because we can only define templates for target attributes that have a clear definition or recurring pattern such as BMI and hypertension. It is not possible to formulate templates for ambiguous target attributes. For example, BioSHaRE researchers manually created the algorithm for the target attribute *Quantity of Beer Consumption* in

LifeLines following the logic 1) whether or not the participants have had any alcoholic drinks (yes/no); 2) if 'yes' the quantity of beer will be returned otherwise a null value will be returned. The pseudo code of the algorithm is shown below,

```
if($('drinking_alcohol').value() == 'yes')
{
        return $('beer_quantity').value();
} else {
        return null;
}
```

However, there are two major remaining challenges in generating this kind of algorithm. First, semantic search is only able to find beer-related attributes; it still misses the alcohol-drinking-related ones because, while subclass relations are used in the query expansion in semantic search, reversed relations are not. The search knows about the fact that **beer** is a *subclass_of* **alcoholic drink** but does not understand that **alcoholic drink** is a *superclass_of* **beer**. We did not include such reversed relations in the query expansion to prevent semantic search from finding too many false positives (irrelevant source attributes). This problem could be solved in the future by including a 'semantic relatedness' metric into the system. Wu and Palmer proposed to calculate the semantic similarities of any two concepts by considering the depths of the concepts within the ontological hierarchy and the lowest common ancestor in the WordNet taxonomy [76],

$$WUP\_similarity = \frac{2 \times depth\_of\_lowest\_common\_ancestor}{depth\_of\_concept^1 + depth\_of\_concept^2}$$

For example, the semantic similarity for 'beer' and 'alcoholic drink' is 91% when using the tool provided by wsj4 Java library online demo http://ws4jdemo.appspot.com/?mode=w&s1=&w1=beer%23n%231&s2=&w2=alcoholic_drink%23n%231 [77].

Second, even if suitable source attributes (**beer** and **alcoholic drinks**) can be found by semantic search, the algorithm generator doesn't know how to handle them because there are no suitable templates for these two attributes. One of potential solutions would be to train the system to learn the patterns of the existing algorithms defined by the human experts, i.e. to reuse all the matches that have been created before as potential templates. This would enable the system to utilize the human expert knowledge now implicitly available in the data conversion algorithms.

Repeated measurements

We observed that the same attribute is often measured multiple times to reach a high precision or to establish time series. For instance, in the Mitchelstown biobank, systolic blood pressure was measured three times: *systolic blood pressure 1st reading*, *systolic blood pressure 2nd reading* and *systolic blood pressure 3rd reading*. When the target attribute *Systolic Blood Pressure* is matched to Mitchelstown, we could decide to take the average value of those source attributes. Because all the repeated attributes are lexically close, it would be possible for the system to check if the top suggested attributes are repeated measurements and then decide whether or not to take the average value.

Matching and recoding of categorical data

To robustly match categories, we not only enabled lexical matching but also developed a new frequency matching method (see **Supplementary Table S6**). Moreover, we introduced a rule-based category matching system in which we have hardcoded rules to make the system smart enough to deal with difficult categories (see **Supplementary Table S7**). Most of the categories shown in the evaluation section could be matched correctly, but there will no doubt be new special cases that require us to add new rules. We would like to allow users to define custom rules for matching categories in the database. For matching string-type data values, we have developed a tool (SORTA) to semi-automatically recode the values based on the selected coding systems or ontologies, which we plan to incorporate in the near future [78].

Statistical matching

Although units are now accurately detected from the label of attributes using the string-matching algorithm, not all attributes actually contain any information regarding units. In those cases, users now have to guess the unit from data values based on their empirical experience. However, when biobank datasets are available in the system, it should be possible to extrapolate the units using a statistical approach in which the distribution of data values is compared to the distributions of other source data values for which unit information is available.

4.6 Conclusion

We have introduced and demonstrated the utility of MOLGENIS/connect, a generic computer system for semi-automatic harmonization and integration of data with focus on human phenotypes in biobanks, patient registries and biomedical research. The system includes a novel method to automatically generate harmonization/integration algorithms based on ontological query expansion, lexical matching and algorithm template matching. Evaluation in 184 BioSHaRE matches showed MOLGENIS/connect is able to generate useful matches and algorithms in 73% of the cases while only 11% still needed to be created by completely hand. Users can use these auto-generated algorithms to rapidly design and execute the integration via a user-friendly online web application. The application and source code are available as open source via the MOLGENIS software suite at http://github.com/molgenis/molgenis and a demo can be found at http://www.molgenis.org/connect.

Acknowledgements

We thank our collaborators in BioSHaRE, Maelstrom Research and OBiBa (most notably Yannick Marcon, Vincent Ferreti), the contributing biobanks (LifeLines, Prevend and Mitchelstown), our collaborators from RD-Connect (most notably Lucia Monaco and the partner biobanks), BBMRI-ERIC (most notably Kaisa Silander, National Institute for Health and Welfare, Helsinki) and Kate Mc Intyre for editing the manuscript.

Funding

This work was supported by the European Union Seventh Framework Programme (FP7/2007-2013) grant number 261433 (Biobank Standardisation and Harmonisation for Research Excellence in the European Union - BioSHaRE-EU), grant number 284209 (BioMedBridges) and grant number 305444 (RD-Connect) and by BBMRI-NL, a research infrastructure financed by the Netherlands Organization for Scientific Research (NWO), grant number 184.021.007.

Chapter 5

BiobankUniverse: automatic matchmaking between datasets with an application to biobank data discovery and integration

Chao Pang[1,2], Fleur Kelpin[1], David van Enckevort[1], Niina Eklund[3], Kaisa Silander[3], Dennis Hendriksen[1], Mark de Haan[1], Jonathan Jetten[1], Tommy de Boer[1], Bart Charbon[1], Petr Holub[4], Hans Hillege[2], Morris Swertz[1,2,*]

[1]University of Groningen, University Medical Center Groningen, Genomics Coordination Center, Department of Genetics, Groningen, the Netherlands. [2]University of Groningen, University Medical Center Groningen, Department of Epidemiology, Groningen, the Netherlands. [3]Genomics and Biomarkers Unit, National Institute for Health and Welfare, Helsinki, Finland. [4]BBMRI-ERIC, Graz, Austria.

***Bioinformatics**, btx478 (August 2017)*

DOI: 10.1093/bioinformatics/btx478

Abstract

Motivation: Biobanks are indispensable for large-scale genetic/epidemiological studies, yet it remains difficult for researchers to determine which biobanks contain data matching their research questions.

Results: To overcome this, we developed a new matching algorithm that identifies pairs of related data elements between biobanks and research variables with high precision and recall. It integrates lexical comparison, Unified Medical Language System ontology tagging and semantic query expansion. The result is BiobankUniverse, a fast matchmaking service for biobanks and researchers. Biobankers upload their data elements and researchers their desired study variables, BiobankUniverse automatically shortlists matching attributes between them. Users can quickly explore matching potential and search for biobanks/data elements matching their research. They can also curate matches and define personalized data universes.

Availability and implementation: BiobankUniverse is available at http://biobankuniverse.com or can be downloaded as part of the open source MOLGENIS suite at http://github.com/molgenis/molgenis.

Contact: m.a.swertz@rug.nl

5.1 Introduction

The increasing breadth and depth of data in the biological sciences provides many new opportunities to understand the mechanisms that underlie complex diseases and essential background for personalized medicine and health. Much of this data resides in biobanks, which not only store sample collections (urine, blood and DNA) but also large data collections (e.g. history of disease, physical activity, lifestyle and environmental factors) (Scholtens et al., 2015). With so many valuable resources available, one would expect much more scientific output for each biobank at an ever-increasing pace.

However, while working on various biobanking projects over the past five years, we noticed limited biobank reuse. What we observed instead was researchers spending a substantial amount of their time locating, negotiating access to and interoperating biobank data before they could actually study the pooled data. There are useful standards emerging for describing biobank collections such as MIABIS (minimum information about biobank information) (Merino-Martinez et al, 2016), directories that list all available biobanks (Holub et al., 2016), catalogues of biobank data schemas (Maelstrom Research, 2015) and robust integration protocols (Fortier et al., 2010). However, researchers still routinely ask us how to find suitable biobank data collections for their research questions. They also spend many months manually curating and comparing biobank data elements to define integrated datasets because existing tools do not enable automatic matching.

In our recent experience the process of data harmonization and integration, driven by a research question, typically consists of the following steps (Fortier et al., 2010): 1) find the datasets relevant to the research question; 2) determine the harmonization potential between the target schema representing the research question and data elements in the relevant dataset; 3) identify the attribute matches between the target schema and the source data for integration. Through a series of user workshops we listed several use cases in **Box 1**, based on which we have identified three major user needs in biobank data discovery:

1. Researchers want to **find biobank data collections** that can be potentially useful in terms of relevant data items in order to shortlist biobanks that might be suitable to serve a particular research project.
2. Researchers want to **assess the integration potential** of data collections and their data items (matching research variables) as the basis for data requests and to make decisions about whether it is worthwhile spending time on data integration for pooled analysis.
3. Biobanks (and networks of biobanks) want to **identify attribute matches between similar biobank data collections** to provide integrated datasets as basis for large studies.

Box 1: Overview of catalogue projects for data discovery

BBMRI-ERIC biobank directory: Main use case is to give an overview of the landscape of biobanks and biobank collections in the BBMRI-ERIC member states.
BBMRI-NL biobank catalogue: Main use case is to advertise all biobank collections available in Netherlands and lead interested researchers to contact these biobanks.
RD-Connect sample catalogue: Main use case is to give a comprehensive overview of the available samples for rare diseases.
LifeLines catalogue: Main use case is to allow the researcher to find and request access to data items of interest.
Maelstrom Research: Main use case is to provide harmonization potential (data attributes) between standard target data schemas and biobank studies.

In addition, all these use cases needed to be served using only metadata descriptions of the data, as individual level data is typically subject to data access committees because of privacy constraints. Joining forces with the BBMRI and ELIXIR infrastructures and the CORBEL, ADOPT and RD-Connect projects, we have developed BiobankUniverse. BiobankUniverse is an online service that bridges the biobank data discovery gap by (a) enabling users to share data element descriptions of biobank data collections and (b) providing a new matching score that identifies pairs of related data elements between biobanks and research variables.

5.2 Methods

In previously published work, we developed BiobankConnect (Pang et al., 2015), a semantic search tool for matching data items between biobank data collections using ontology-based query expansion on top of the information retrieval system Lucene (The Apache Software Foundation, 2006). However, while achieving high precision and recall, BiobankConnect still requires substantial user input. Specifically, each of the desired 'target' attributes needs to be manually annotated with ontology terms before the system can try and find relevant 'source' attributes from biobanks that match this target. This is only feasible if the user wants to compare many 'source' biobanks against one relatively small 'target' set of data items.

To enable pairwise discovery considering all data items of many biobanks without requiring extensive curation we have developed a new algorithm that automatically shortlists matching data items between any two or more collections of data elements (such as data schemas in biobanks). To standardize the terminology throughout this paper, we will use 'attribute' to refer to a variable, data column, data element or data item. We implemented the algorithm as open source in Java and reused data management tools and user interfaces from the MOLGENIS software platform (Swertz et al., 2010).

Figure 1 provides an outline of the system, which consists of six key steps: 1) automatic ontology tagging of attributes using lexical matching, 2) matching pairs of attributes using ontology-based query expansion, 3) matching pairs of attributes using lexical matching, 4) prioritizing matches from both lists by calculating a normalized similarity score, 5) filtering irrelevant matches based on key-concepts to improve precision, and 6) calculating semantic similarity scores between biobank pairs. Each step is described in detail below.

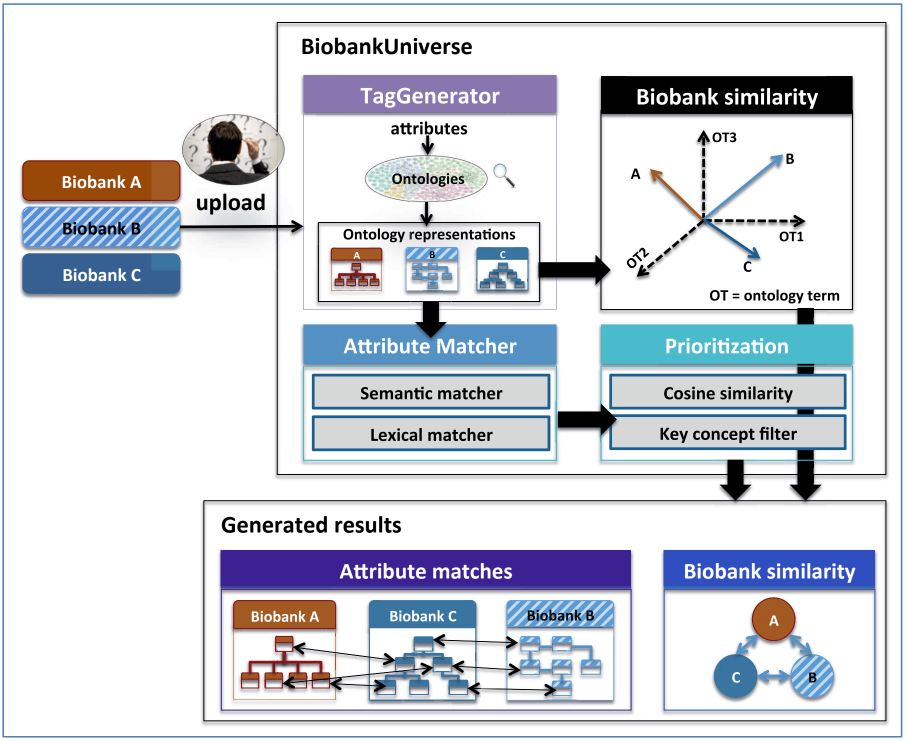


Figure 1 | **The overall of the BiobankUniverse system.** Users upload/add biobanks attributes to the universe. TagGenerator is automatically triggered to create ontology representations of the uploaded biobank's attributes. These are then used in AttributeMatcher to generate attribute matches with any of the other biobanks. A cosine similarity score is computed for each attribute match pair to prioritize the candidate list, and a strict matching criterion is applied to remove false positives. A biobank similarity is also calculated by computing the cosine angles between the ontology representations of biobanks in the semantic space for each pair.

Automatic ontology tagging of attributes using lexical matching

Because of their heterogeneous backgrounds, biobanks often describe their attributes using very different terminologies, which hinders the automatic matching of related or equivalent attributes. To enable matching based on these heterogeneous metadata, we 'tag' each attribute with one or more groups of ontology terms based on the label + description. For example, 'History of Hypertension' is tagged with two groups of ontology terms: (History && Hypertension) and (Medical history [synonym: History] && Hypertension). Each group of ontology terms is called a tag group.

With BiobankConnect, users had to do this tagging manually, which was not feasible when matching dozens of biobanks with thousands of attributes. In BiobankUniverse, each attribute is tagged automatically in four steps: 1)

Having indexed the Unified Medical Language System (UMLS) ontology (UMLS is a meta-thesaurus that incorporates all major biomedical ontologies such as SNOMED CT, NCI thesaurus and ICD-10), we use the Vector Space Model (VSM) to find potentially relevant ontology terms for each attribute based on its label; 2) We apply a strict matching criterion to remove non-informative ontology terms. Only ontology terms (or synonyms) whose labels (or any their synonyms) can be completely matched to words from the attribute label are considered as tags; 3) We use a cosine-similarity-based string-matching algorithm to compute a similarity score between the attribute and the ontology terms, which we use to order the tags from most relevant to least relevant; 4) We remove non-informative tags. In this step, we use ontology terms with the highest similarity as the initial tag group then prune the rest of the list to see if inclusion of the next ontology terms as the tag group results in an overall improvement of the similarity score. If yes, we keep the new ontology term in the tag group. If no, we remove the term and repeat the same procedure for the next item in the list. The result is a set of ontology term tag groups for each attribute. An example of tagging attribute is shown in **Supplementary example S14**. In (Pang et al., 2015), we discussed how to select ontologies for this procedure based on the extent that an ontology covers the data. Based on these experiences, we decided to use UMLS.

Matching pairs of attributes using ontology based query expansion

The tags established in the step (**Automatic ontology tagging of attributes using lexical matching**) are now used to search for semantically matching pairs of attributes between biobanks using semantic query expansion in a manner similar to what we previously described for BiobankConnect (Pang et al., 2015). We have now changed the algorithm to query on terms from both parent and child classes (instead of child only) to ensure that the matches generated by this query expansion are symmetrical. This ensures that queries of more specific biobank attributes will still find matching attributes from another biobank that are tagged with more general ontology terms. An example of matching attributes is provided in **Supplementary example S15**.

In BiobankUniverse, we have also optimized query execution. In BiobankConnect, we created separate queries for each attribute to match a small number of attributes (<100). This is computationally too expensive for large numbers of biobanks with large numbers of attributes because we have encountered many attribute-matching cases, where more than 100,000 of expanded queries needed to be collected from the UMLS ontology and this process dramatically slowed down the matching process. Thus, in BiobankUniverse, we implemented a more efficient matcher that uses the hierarchical ontology term relations to discover the matching correspondences between those attributes. For example, the concept ‘Vegetables’ is a parent class of the concept ‘Beans’ so inferentially the attributes tagged with ‘Vegetables’ can be concluded as the matches for the attributes tagged with ‘Beans’.

To efficiently compare these hierarchical relationships, we collect all the term paths available for the tagged ontology terms into a list of atom unique identifiers of the current concept and its ancestors. For each attribute, we then check whether this term path or any of its parent term paths overlaps and, if so, we retrieve the corresponding attributes as the candidate match.

For example, the attribute ‘Consumption of Vegetables’ has path ‘A3684559.A3206010.A3314529.A2881738.A3217489.A2887927’ and the attribute ‘Consumption of Beans’ has overlapping path ‘A3684559.A3206010.A3314529.A2881738.A3217489.A2887927.A3189886. A2878987’, so we can conclude that ‘Consumption of Beans’ is a more specific match for ‘Consumption of Vegetables’ based on their paths. To prevent false positive matches based on very general concepts, we decided to limit the upward traversals to stop at level 5 from the root of UMLS after evaluating different cut-offs as discussed in section 5.4.

Matching pairs of attributes using lexical matching

We also implemented a lexical matcher that uses standard search functionality from ElasticSearch. Given an attribute label/description from one biobank, the lexical matcher retrieves attributes from another biobank that share at least one word (excluding punctuation marks and stop words). The purpose of this matcher is to retrieve matches where the attribute labels are very similar and to retrieve attributes that have no tags to use for semantic matches. The motivation for this second method is that some of the attributes use terminology not yet defined in any ontology such as the attribute 'SOKRAS sticker series' in Finrisk2002 and Finrisk2007. Enabling lexical matching will help capture the matches containing those specific attributes.

Calculating a normalized similarity score to prioritize matches from both lists

The previous two steps (**Matching pairs of attributes using ontology based query expansion** and **Matching pairs of attributes using lexical matching**) produce two lists of candidate matches for each attribute based on the lexical matcher and the semantic matcher, respectively. To merge both lists, we calculate a similarity score for each matching pair using the cosine similarity algorithm also used in Lucene [45]. In this score, each 'query' attribute from one biobank and its candidate matches from another biobank are treated as vectors in a space built of all words derived from all attribute names and descriptions. For each vector, the length of the dimension (word) is calculated by multiplying the word inverse document frequency with the word occurrence in the specific attribute. The vector and similarity score are computed as:

$$\overrightarrow{Vector} = (Word\_1_{tf} \times Word\_1_{idf}, \ldots, Word\_n_{tf} \times Word\_n_{idf})$$

$$Cosine = \frac{\sum_{i=1}^{n} Vector\_target_i \times Vector\_candidate_i}{\sqrt{\sum_{i=1}^{n} Vector\_target_i{}^2} \times \sqrt{\sum_{i=1}^{n} Vector\_candidate_i{}^2}}$$

It was particularly complicated to generate meaningful scores in cases where a pair of attributes are semantically close but have very different labels. This results in very low cosine similarity scores for matches that an expert user would recognize as a good match, e.g. 'Consumption of Vegetables' vs. 'Consumption of Beans'. We therefore also calculate a cosine similarity score based on the ontology terms instead of the attribute labels.

For each pair of attributes, we first retrieve all ontology tags that are either the same or related via parent-child or child-parent. We then replace the relevant substrings of the attribute labels with information from their ontology tags. For example, ‘History of high blood pressure’ and ‘History of hypertension’ are converted to ‘History of hypertension’.

If ontology terms are related via a parent-child or a child-parent relationship, we replace the child ontology terms with the parent terms in the attribute labels. However, these parent/child ontology terms are obviously not equivalent with the attribute label, just of a sub/superclass. We therefore correct their similarity score based on the semantic-relatedness between these parent and child ontology terms [76]. This correction is only performed on the subscore that is contributed by the relevant substring replaced by the information from ontology tags as follows:

$$Relatedness = \frac{Level_{parent} \times 2}{Level_{child} + Level_{parent}}$$

$$Score_{sub} = Score_{total} * \frac{Length_{replacement}}{Length_{total}}$$

$$Score_{corrected} = Score_{total} - Score_{sub} + Score_{sub} \times Relatedness^2$$

For example, when calculating the similarity score between attribute ‘Consumption of Vegetables’ and attribute ‘Consumption of Beans’, ‘Beans’ (level 8) is replaced with more general term ‘Vegetables’ (level 6). Without correction, the cosine similarity score would be 100% because both attribute labels are the same, which is clearly too high a score because the attributes are of semantically different levels. To correct for this, we first of all calculate the relatedness between ‘Vegetables’ and ‘Beans’,

$$Relatedness = \frac{6 \times 2}{6 + 8} = 0.857$$

We then calculate the subscore that is contributed by ‘Vegetables’,

$$Score_{sub} = 100\% * \frac{10}{23} = 43\%$$

Finally we compute the corrected score,

$$Score_{corrected} = 100\% - 43\% + 43\% \times 0.857^2 = 88.6\%$$

After we have calculated all the similarity scores for all the candidate attribute matches, we sort the list based on similarity scores and keep (at most) the first 50 matching pairs (50 is the limit of user-acceptable matches based on BiobankConnect user feedback). [50]

Filter out irrelevant matches based on key concepts to improve precision

The BiobankUniverse search methods are optimized to yield maximum recall. However, not all ontology terms are equally relevant for the research domain, and some may yield false positive matches. To reduce false positives, we enable users to filter results to matches that are based on ‘key concept’ ontology terms such as ‘Hypertension’ while discarding more general ontology terms such as ‘History’. For this we use the 'semantic type' of UMLS ontology terms that indirectly indicate the importance of these concepts. For example, ontology terms associated with the semantic type ‘Disease or Syndrome’ (e.g. Myocardial infarction) are key concepts while the semantic type ‘Quantitative Concept’ (e.g. Numbers) indicates the common concepts. We used this as basis for the definition of the key concepts and went through the list of all 127 semantic types in UMLS and manually allocated them to the group of key concepts and the group of common concepts that are used in the system to determine the quality of the matched source attributes. Group members of the semantic types can be found in **Supplementary Table S16**.

Using these key concepts, we apply a lexical matching filter in which all the words from the key concept must be perfectly matched (considering lexical matching methods that allow for stemming etc.). For example, ‘Have you ever had high blood pressure?’ is a good match for ‘history of hypertension’ because both of the attributes are matched on the key concept **hypertension** whereas ‘history of myocardial infarction’ is far less relevant for ‘history of hypertension’ because the matched word **history** is not a key concept.

As an additional filter, attributes need to be matched based on words that are not stop words and consist of at least three alphabetic characters. If these two criteria are not met, the matches are treated as false positives and removed from the candidate list.

Calculate overall semantic similarity between biobanks

Finally, we created a metric to quantify the similarity between two biobank collections. At first we simply calculated the average of the attribute similarity for all of the candidate matches. However, this metric showed bias towards collections that were lexically similar and penalized semantic similarity. For example, the scores of the matches generated between FINRISK2002 and FINRISK2007 are systematically higher than the ones between HOP and Lifelines because FINRISK2002 and FINRISK2007 use very similar attribute labels and descriptions (see description of these biobanks below in the Results section). We therefore implemented a metric that uses the semantic tags of the attributes.

Our new metric compares vectors of unique ontology terms derived from the tags of all attributes of both biobanks. Exactly matching terms are given a value of '1'. Indirectly matching terms (i.e. a parent/child terms) are given a lesser score based on the semantic relatedness [76,77]. Finally, a cosine similarity is calculated on the vectors for the each biobank pair as described above in the previous step (**Calculating a normalized similarity score to prioritize matches from both lists**). For example, Biobank A has attributes tagged with the ontology term 'Vegetables' and biobank B has attributes tagged with the ontology terms 'Beans' and 'Tomatoes'. When combined, there are three dimensions in their space and the vector representations are:

$$\overrightarrow{Biobank\ A} = (Vegetables\colon 1, Beans\colon 0.8, Tomatoes\colon 0.8)$$

$$\overrightarrow{Biobank\ B} = (Vegetables\colon 0.8, Beans\colon 1, Tomatoes\colon 1)$$

The cosine similarity between them is 0.978. Based on this measure, we can generate a matrix containing all pairwise similarities between all biobank collections available. We then visualize the matrix in a network using the Vis 3D JavaScript library to provide users with a visual representation of which biobank collections are closest to each other (see Results section).

5.3 Implementation

We have made the biobank matchmaker algorithm available in a user-friendly web application (http://www.biobankuniverse.org). It can be also downloaded as part of MOLGENIS (http://www.molgenis.org). It uses a domain model (see the file **data_model.pdf in Supplementary material)** that extends the MIABIS standard model for 'Biobank' and 'SampleCollection' description [10]. The system works as follows:

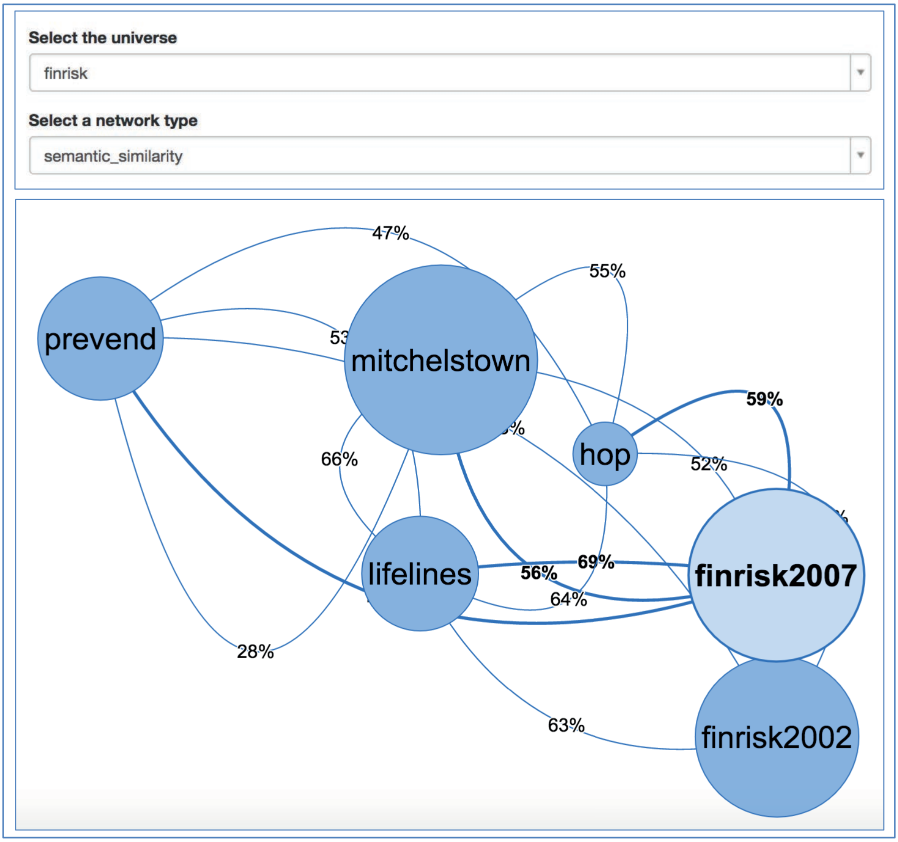
Biobankers upload collection metadata and match their attributes

Biobankers can upload data collection descriptions, i.e., the list of data items of an existing biobank or study for which data items can be shared via CSV. An example file can be found in **Supplementary material prevend_biobank.csv**. At upload, each attribute is automatically tagged with ontology terms. The tag groups and their quality measures (cosine similarity and matched words) are stored in the database for fast retrieval. The software then generates a list of candidate matches for each of the previously loaded biobanks. For example, the attribute ‘Have you ever had high blood pressure’ is matched with the tag group (Hypertension), a record of explanation is as follows, query string = ‘high blood pressure’; matched words = ‘high blood pressure’; ontology terms = ‘Hypertension’; cosine similarity = 50%. All of the information on the matched source attributes, cosine similarities and matched words are stored in the AttributeMappingCandidate table. The tag groups cannot be edited at the moment but will be in the future.

Finding matching biobanks

Researchers and other prospective biobank users can use the system to find biobanks with relevant data and can explore the matching relationships between those attributes using a data discovery user interface (shown in **Figure 2**).

When the page is first loaded, a biobank “universe” is shown in the center of the page beneath the search box. The circles represent biobank members of the universe. The size of the circle indicates the number of attributes the biobanks contains. The connecting lines between circles represent the number of matching attributes between biobank members. Users can define their own queries in the search box at the top of the page. In order to retrieve attributes with high precision, the search box is equipped with an auto-complete function that provides suggestions from the UMLS ontology. Depending on the filter, the biobank universe will be reduced in size and the circles and number of matches will change dynamically. Users can also display the universe showing only human curated matches or using the semantic similarities between biobanks, as described above.



Select the universe

finrisk

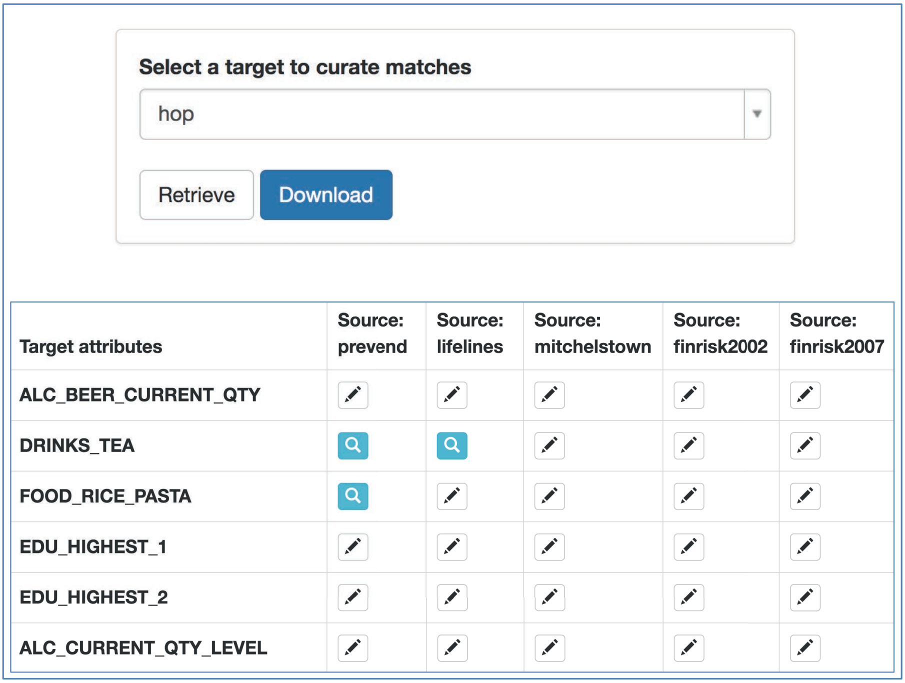
Select a network type

semantic_similarity

Figure 2 | **User interface for discovering biobanks.** Users can choose various network options to visualize the ‘universe’: the biobank similarity, the number of matches generated by the system or the number of matches curated by the user. The nodes represent biobanks in the universe and their sizes are proportional to the number of attributes in the corresponding biobanks. The connecting lines represent the similarities (defined as the number of matches or the biobank similarities) between biobanks, the more similar they are and the closer they are next to each other in the universe. The online version is dynamic so you can see the numbers more clearly.

Exploring and curating attribute matches

Users can drill down to view and compare the attribute matches for a subset of biobanks. To start a comparison session, users first choose one of the biobanks as the ‘target’. For each of its attributes, matches available in the other biobanks are then shown (see **Figure 3)**. Users can manually curate these matches using an editing interface in which they can select or reject matches. To more efficiently curate the large number of matches, we have introduced a batch acceptance feature that enables users to accept/reject all matches at once based on a quality criterion.



Select a target to curate matches

hop

Target attributes	Source: prevend	Source: lifelines	Source: mitchelstown	Source: finrisk2002	Source: finrisk2007
ALC_BEER_CURRENT_QTY	✎	✎	✎	✎	✎
DRINKS_TEA	🔍	🔍	✎	✎	✎
FOOD_RICE_PASTA	🔍	✎	✎	✎	✎
EDU_HIGHEST_1	✎	✎	✎	✎	✎
EDU_HIGHEST_2	✎	✎	✎	✎	✎
ALC_CURRENT_QTY_LEVEL	✎	✎	✎	✎	✎

Figure 3 | **Curating candidate matches by data owners.** Users can curate all generated matches available in the universe. Users first choose a leading ‘target’, based on which a match table is generated. (Any biobanks can be a target because of the pairwise match). Users then need to go through each of the cells in the table to make decisions about the generated matches.

Searching for research variables

One of the main challenges in biobank research is finding datasets suitable for a particular analysis or for testing a particular hypothesis. To speed up this discovery process, users can also upload a complete list of desired research attributes and then start a data discovery job. This list is then shown as an additional circle within the universe. This search interface then works in the same way as the matching curation interface, enabling curation of the matches between desired research variables and biobank data items. The results can be downloaded for use as the basis for a data request.

5.4 Results

The main goal of BiobankUniverse is automatic generation of high quality lists of matching attributes between biobanks. To evaluate precision and recall, we re-ran our evaluation procedure from BiobankConnect [50], which compares automatically found matches against human curated (relevant or 'correct') matches as follows:

$$Recall = \frac{\#Found\_relevant\_matches}{\#All\_relevant\_matches}$$

$$Precision = \frac{\#Relevant\_found\_matches}{\#All\_found\_matches}$$

We applied this to a new version of the validation data we used in Molgenis/Connect [79]: a human-curated matching set from the BioSHaRE Healthy Obese Project (HOP) consisting of 92 target attributes in three different biobanks [46]. In addition, we also used a curation set between two large biobank collections from the FINRISK project.

BioSHaRE Healthy Object Project performance

We evaluated BiobankUniverse's performance using the complete set of HOP, which consists of 92 target attributes, and three sets of biobank attributes (from the LifeLines, Mitchelstown and Prevend biobanks). There are 66,884 possible matches, out of which 633 were classified as relevant. We observed new average precisions and recalls over ranks ranging from 1st, to 50th (see **Table 1)** that are better than those of BiobankConnect (see **Table 1**) while providing major user time- and cost-savings because substantial manual tagging is no longer required. In addition, the new matching algorithm is more efficient than that of BiobankConnect. It took 2 minutes on average for BiobankUniverse to generate candidate matches between HOP and any of the biobanks, while 1 and half hour approximately for BiobankConnect to generate the candidate matches for the same pair.

Table 1. Recall and precision performance for the HOP project (0-100)

	Lifelines		Mitchelstown		Prevend		Total		Biobank Connect	
Rank	R	P	R	P	R	P	R	P	R	P
1	23	64	23	87	39	41	25	66	24	58
2	39	55	33	66	61	38	38	55	37	45
3	45	45	42	58	70	34	46	47	45	39
4	52	41	48	52	71	32	52	44	50	35
5	56	38	56	50	73	30	58	42	54	32
6	59	35	58	46	74	30	60	39	57	30
7	64	34	62	44	74	29	64	37	60	29
8	66	32	66	43	74	28	67	36	63	27
9	68	30	69	42	77	29	69	35	65	26
10	70	29	72	41	77	29	71	34	67	25
20	85	25	81	36	77	28	82	30	76	19
50	88	20	85	34	77	28	85	26	77	16

P, precision; R, recall.

FINRISK large collection matching performance

We also evaluated the performance of BiobankUniverse using the National FINRISK Study, survey years 2002 and 2007, which involved matching two large biobank collections against each other with potentially 581,742 possible matches (798*729), of which 550 of were classified as 'correct' by human curators. Although the two surveys were conducted by the same research group, they were created in different time periods and the questions asked changed over time, thus requiring this integration effort. The motivation for matching these two collections is that they are often used together in analyses.

For example, the attribute ‘Siblings diagnosed with asthma’ collected in FINRISK 2002 changed to ‘sisters diagnosed with asthma’ and ‘brothers diagnosed with asthma’ in FINRISK 2007. Researchers who want to use data from both of the collections usually need to match the two sets of attributes with each other manually. In order to manually match all attributes in these two collections, the FINRISK researchers performed the following process: they organized and tabulated all attributes into topics one study at a time, and then compared the attributes against the items in the other collection, first inside each topic and then across the full collection if no match was found inside a topic. The quality of the matches was scored using SKOS mapping system [80]. The full tabulation and comparison of the two collections was labor-intensive, taking approximately 2 working days. It is important to note that this work was done by a person highly familiar with these collections – the work would have taken longer for someone not familiar with them.

Table 2. Recall and precision performance for the FINRISK project (including 550 manual matches)

Rank	Recall	Precision	Retrieved
1	0.813	0.592	755
2	0.878	0.325	1486
3	0.891	0.223	2197
4	0.898	0.171	2889
5	0.904	0.139	3563
6	0.911	0.119	4214
7	0.913	0.104	4834
8	0.915	0.092	5438
9	0.918	0.084	6032
10	0.922	0.077	6614
20	0.929	0.044	11605
50	0.938	0.027	19088

We applied BiobankUniverse to FINRISK 2002 and FINRISK 2007 tabulated attributes and generated a set of matches between them. These matches were compared to the manually created list of matches (see **Supplementary material FINRISK2002-FINRISK2007-relevant-matches.xlsx)**. We computed precision and recall using the procedure described above, and found a recall of 0.81 precision of 0.59 at rank 1st and recalls of 0.92, 0.93 and 0.94 at rank 10th, rank 20th and rank 50th respectively, the complete set can be found in **Table 2**. According to the FINRISK researchers, approximately identifying a correct match within the top 10 candidate matches takes 10–20 seconds (ignore candidates outside the top 10). The complete curation process for 800 pairs of matches would take about 2–4.5 hours and identify 92% of the true matches.

5.5 Discussion

Below we discuss improvements over BiobankConnect, how to reduce false positives, potential improvements of the matching procedure beyond lexical and semantic matching and other future work.

Improvements over BiobankConnect

BiobankUniverse is the successor to BiobankConnect, which was developed to find matches between a small target schema describing variables for a research project and large biobank schemas that (hopefully) provide these variables. BiobankConnect, however, required an unacceptable level of user interaction to achieve matching results with high precision. In BiobankUniverse, we therefore worked to reduce manual effort as much as possible. First, we enhanced automatic tagging to capture as many tag groups as possible. Second, we used UMLS semantic types to automatically remove false positives. Third, we introduced an objective measure to calculate the cosine similarity score and to discover matched words in order to provide users with a fairly good idea how the matches were generated. All together, these improvements enabled us to match large biobank collections against each other, and it is very encouraging to see that BiobankUniverse performs similarly to the more human-labor-intensive BiobankConnect.

Use of strict matching criteria to reduce false positives

Users questioned the added value of filtering using key-concepts. In response, we compared recall, precision and the number of matches retrieved with and without this filter using the HOP project data (see **Table 3** for results). Applying the key-concept filters resulted in many fewer candidate matches while systematically increasing recall and precision. This is exactly as desired because the main purpose of these criteria is to improve precision by removing false positives so that users need to review fewer invalid candidate matches before finding all relevant matches.

Table 3. The overall performance comparison while enabling and disabling the matching criteria from the HOP experiment (including 633 manual matches)

	Matching criteria enabled			Matching criteria disabled		
Rank	R	P	RE	R	P	RE
1	0.25	0.66	240	0.24	0.56	268
2	0.38	0.55	443	0.36	0.44	516
3	0.46	0.47	613	0.43	0.37	735
4	0.52	0.44	753	0.50	0.34	931
5	0.58	0.42	877	0.54	0.31	1089
6	0.60	0.39	987	0.58	0.30	1235
7	0.64	0.37	1085	0.61	0.28	1373
8	0.67	0.36	1173	0.63	0.26	1506
9	0.69	0.35	1250	0.65	0.25	1630
10	0.71	0.34	1320	0.68	0.25	1751
20	0.82	0.30	1724	0.76	0.18	2723
50	0.85	0.26	2054	0.80	0.13	3848

P, precision; R, recall; RE, number of retrieved matches;

As shown in the examples in **Table 3**, users had to check 431 (1751-1320), 999 (2723–1724) and 1794 (3848-2054) fewer matches when applying the strict matching criteria at rank 10th, 20th and 50th. Suppose that rejecting a false positive would take a minimum of 10 seconds (in reality it could be more), users would have to spend at least 1, 3 and 5 hours more to curate candidate matches at rank 10th, 20th and 50th respectively.

Improving ontology coverage of the domain

We could account for some of the poorer attribute matches because they were based on attribute labels from HOP that don't exist in the UMLS ontology, for which the system consequently couldn't use semantic matching. For example, the target attribute 'Current Consumption Frequency of Bakery Products' is manually matched to eight source attributes (e.g. Pancakes, Fruit Pies) in Mitchelstown, but the system failed to retrieve any of the relevant attributes. We know, retrospectively, that if the concept 'Bakery Products' had been annotated with the ontology term 'Starchy food' then all of the relevant matches would have been found by the system because all eight matches have been annotated with the ontology terms that are the subclasses of 'Starchy food' (e.g. Pancake is a descendant of Starchy Food).

Limiting the query expansion in the parent direction

During the development of BiobankUniverse, we realized that expanding queries towards the parent direction might result in unexpected matches as these include very broad concepts such as Disease or Food. We therefore experimented with various heuristics to remove these matches. The most promising results were achieved by limiting the distance from the root of the ontology at which the query expansion would stop. We therefore calculated recall and precision using the HOP data for 1-6 levels from the root (results shown in **Supplementary Table S17**). What we found was that precision increased with level up to level 5 from the root. This is because concepts are less general at higher levels and thus fewer false positives are produced. However, precision started to decline beyond the level 6. We also found that recall was relatively steady from the root up to level 5, then started to drop at the level 6. Apparently level 6 contains some informative ontology terms that help in the semantic matching. More importantly, the level 5 cut-off produces the best f-measure compared to other levels, we therefore chose level 5 as the final cut-off.

The limitation of the lexical and semantic based matching algorithms

The use of ontologies in matching algorithms has been effective in matching attributes, especially in resolving the differences between datasets in case of synonyms, hypernyms and hyponyms [50]. However, we still often encounter difficult cases where the attribute is described in a non-standard way and ambiguously. For example, the LifeLines attribute FOOD7A1 ‘How many cups did you on average use on such a day?’ should be matched to the target attribute ‘Current Consumption Quantity Of Coffee’. In this case the source attribute doesn’t have any mention of ‘Coffee’ in the description and it’s not clear that the question is referred to coffee, tea or something else. Thus only humans having inside knowledge are able to find such attribute matches.

We have piloted technical solutions for such ambiguities. For instance, we can use the language model GloVe, which is an unsupervised learning algorithm for obtaining the vector representations for words [81]. The trained GloVe model outputs the probability for the word pair that indicates the likelihood of its co-occurrence. In the previous example of matching the key word ‘tea’ to ‘coffee, we could use the GloVe model to find a list of the most frequently co-occurred words for ‘coffee. Because ‘cup’ and ‘coffee tend to appear quite often, we should see the word ‘cup’ ended up in the list and hence be able to succeed in matching ‘Current Consumption Quantity Of Coffee’ to ‘How many cups did you on average use on such a day?’. We envision use of such technologies to further improve the matching algorithm.

Future perspectives for BiobankUniverse

Currently BiobankUniverse is used as a mapping tool where users can generate, curate and download the attribute matches. Our ultimate goal is to have a community powered service where everybody can submit their data dictionary to the existing 'universe'. The use case doesn't need to be restricted to the biobank domain only. We envision that other universes can be created using the same toolset. Currently we ask collaborators to send us data collections for uploading but plan to provide comprehensive documentation and video trainings for data contributors to enable self-service. We also want to start collaborations with registries such as EU directory (containing 500+

collections) to incorporate more data collection metadata [4]. Additionally we encourage not only data owners but also researchers to identify matches between datasets to improve the quality of the universe. BiobankUniverse will be particularly useful for discovering relevant datasets by searching certain combinations of selection criteria (certain ontology concepts) and determine harmonization potentials by quickly uploading their own data schema to find data sources in the universe. We realize we need to develop more advanced user interface components to accommodate these advanced use cases. For example, we plan to add more details about attribute matches in the universe for users to interact with. Finally we must invest in performance. In the current system it takes approximately 20 minutes for a laptop with a 4 core CPU and 8 GB RAM to generate matches between one pair of biobanks each containing 1000 attributes. In a biobank universe with 10 members, we would need to calculate 45 pairs. If all these biobanks also contain 1000 attributes, it would take 15 hours to construct the universe. As the universe grows, the computation time will grow near exponentially {time=N*(N-1)/2}. To address this problem, we plan to implement a more scalable pipeline to generate matches that can farm the matching across a parallel computer cluster.

5.6 Conclusion

We have created the BiobankUniverse system for quickly matching data attributes between biobanks by fully automating the matching procedure and by providing new user interfaces for data discovery and matchmaking. While saving much time and eliminating handwork, the performance of the system is also improved compared to the previous system BiobankConnect. In conclusion, we not only increased the speed of the system but also in the mean time we managed to maintain and improve the quality of the candidate matches.

Acknowledgements

We thank the communities of MOLGENIS, CORBEL, BBMRI-NL, BBMRI-ERIC, ELIXIR and RD-Connect for assistance in the software development; THL Biobank for the FINRISK test case (to KS, NE); and BioSHARE for the HOP test case; and Kate Mc Intyre for editing the manuscript.

Chapter 6

Discussion

Pooled data analyses play a crucial role in uncovering subtle associations between phenotypes and complex or rare diseases. By integrating data from multiple data repositories we can obtain a much larger sample size for our analysis with sufficient statistical power to gain more insights into, e.g., the mechanism of the disease development. However, the major barriers to pooled data analyses are difficulties discovering relevant datasets or data elements (data discovery), difficulties mapping heterogeneous data to a comparable standard (data harmonization), and finally difficulties pooling data into one dataset that is ready for analysis (data integration). This thesis has developed novel computational methods and suitable implementations to efficiently resolve the differences in data capture and description among data repositories so that researchers can efficiently discover relevant data and then harmonize and integrate this data for pooled analyses. Below, we discuss our main results, put these results into context with related developments, and address future perspectives.

6.1 Summarizing discussion

In the introduction chapter, we formulated research questions that focus on three different challenges in data harmonization, discovery and integration: 1) semantic ambiguity caused by differences in the metadata or data elements of various data sources, 2) the use of non-standard coding systems to encode data values and 3) the existence of proxy equivalent measurements for the same construct requiring data transformation algorithms to make these data comparable for analysis.

These challenges make reuse of biobank data, in particular pooled analysis of biobank data, very labour-intensive and time-consuming. This is because data harmonization is a complex cognitive task that consists of finding data elements that are either lexically or semantically similar and defining a data conversion algorithm, if one exists, that makes the data values comparable as the basis for integrated analysis. While solving one search task is easy, humans are not able to perform thousands of such cognitive repetitive tasks with great efficiency. Computers generally accomplish such iterative tedious task with great ease but are not equipped to do complex cognitive tasks. What this shows is that humans and computers happen to complement each other's shortcomings.

We therefore hypothesized that computational methods and suitable software implementations that can assist users in automatically resolving differences will be a breakthrough for data harmonization and integration, and thus increase dataset reuse. The methods developed, and their impact on the data integration process, are described below.

Ontology based method for harmonization of semantic ambiguity

In chapter 2 we introduced BiobankConnect [82], a method which combines computational methods for ontology-based query expansion with the information retrieval engine Lucene to shortlist and prioritize candidate matches from biobanks for target data elements of interest. Although the system is able to detect some proper ontology terms for the target attributes, it works best when users manually provide precise ontology term annotations, and users sometimes need to try different ontology terms to find the relevant matches. There are two major sources of added value in BiobankConnect: 1) it automatically expands queries by using synonyms and child classes of the annotated ontology terms, and 2) it takes the information content into account so the order of the candidate matches is more likely to prioritize the true matches. The downside of the tool is that it still requires a lot of user interaction that doesn't scale up for large target schema.

In chapter 5 we introduced BiobankUniverse, an enhanced version of BiobankConnect. BiobankUniverse adds a new computational method to automatically tag attributes with ontology terms using Unified Medical Language System (UMLS), guaranteeing that the correct ontology term annotations are captured (and ultimately outperforming the human curators performing the same cognitive task manually in BiobankConnect). In addition, we developed a new scoring algorithm for BiobankUniverse that objectively computes the similarity between attribute matches even if the pair is lexically very different (e.g. “vegetables” vs. ”beans”). Finally, BiobankUniverse includes a visualization of biobank similarities computed using the semantic similarity, the number of candidate matches and the number of curated matches.

In both BiobankConnect and BiobankUniverse, we pioneered the use of ontologies in combination with information retrieval techniques to match metadata from large biobanks. Notably, we automated the complete matching procedure in BiobankUniverse so that no human input is required to generate attribute matches while also maintaining equivalent performance in recall and precision compared to BiobankConnect. With this automation we have opened up the possibility of scaling the matching process up to hundreds of biobanks. A task that approaches unfeasibility when performed manually.

At present the BiobankUniverse tagging algorithm is optimized to capture as many relevant ontology terms as possible (including the false positives), hence inevitably generating the wrong attribute matches in the subsequent step. We next aim to improve the tagging algorithm by using Natural Language Processing tools [83], which will be discussed in detail in section 6.4 below.

Harmonization of non-standard coding systems in data values

In chapter 3 we introduced SORTA [78], a tool that provides computational methods to (semi-)automatically recode free text data to ontology terms, which is implemented in a wizard-like web application. It uses ElasticSearch, an advanced full text search engine, for fast retrieval of the relevant ontology terms, then calculates the lexical similarities between the text values and the candidate matches using an n-gram-based string-matching algorithm. Users can then define a quality threshold to automatically accept candidate matches with confidence, i.e. they can let the system automatically convert the free text values into a systematic code system if the match is above a set threshold. SORTA also includes a built-in learning mechanism to gain knowledge when a user manually curates low quality matches such that these manual matches are used to automatically re-evaluate the remaining recoding tasks.

SORTA outperformed other similar tools (BioPortal and ZOOMA), especially with respect to recall because we optimized the system to retrieve as many potential matches as possible. In addition, in SORTA, we provide objective scores ranging from 0-100% for candidate matches to help users quickly make their final match selections. An example of SORTA's power is our application of the program to recode 90,000 data entries in LifeLines. In the initial SORTA recoding run 60% of data could be successfully recoded at rank 1 and 80% fell within rank 10, resulting in recalls/precisions of 0.59/0.65 at rank 1 and 0.80/0.14 at rank 10. Further, thanks to the learning mechanism, the second round of recoding the same dataset (or a similar dataset) resulted in improved recall and precision of 0.97/0.98 at rank 1.

At the moment, SORTA users need to specify a good quality threshold to automatically accept the candidate matches. Unfortunately, it is hard to know this quality value threshold beforehand, making the threshold more useful for future recoding tasks on similar datasets. Beside the similarity score, we also need to use a multi-factor system to explore the dynamic threshold. For example, when the candidate score is much higher than the rest of the candidate scores, it is probably safe to assume that the first candidate is true. In addition we could take coding system structures into account in the matching algorithm. Because the algorithm produces the candidate matches that are lexically similar to the input data value, those candidate matches are similar to each other as well. These candidates (ontology terms/codes) therefore tend to originate from "clusters", which can be considered as topics, with the correct match normally belonging to the mostly likely cluster (defined as the cluster including the most candidate matches as members).

Harmonization of data values for proxy equivalent measurements

In chapter 4 we introduced MOLGENIS/connect [79], a system that provides computational methods to automatically generate transformation algorithms that convert source data to the target standards (i.e. addressing the harmonization challenge). Using MOLGENIS/connect, users can semi-automatically integrate large biobank and clinical data into one unified view. Integration of BiobankConnect allows the system to use automatically generated matches between source and target data items as an ingredient for creating algorithms. MOLGENIS/connect then adds the algorithm generator, which includes an automatic unit converter and a category mapper to ‘guess’ the correct algorithm. In addition, the system allows users to define new templates for generating algorithms, e.g. BMI = weight (kg)/height (m)^2. Once all the algorithms are finalized, the system executes the algorithms in participating biobanks to create an integrated dataset.

The combination of BiobankConnect and the algorithm generator allows users to modify candidate matches and algorithms continuously and efficiently, providing much flexibility. Because the algorithm generator uses candidate matches from either BiobankConnect or the user’s selections as the input, users can decide which candidates should be incorporated to give the generator a better chance of producing high quality algorithms. In our experiment, 73% of the results were considered useful: 27.7% of the algorithms were generated perfectly without any curation, 16.8% were generated correctly after the correct source attributes were provided, 16.8% were partially correct and required slight user modification, and 11.7% were generated with user assistance in the selection of matched attribute as well as the modification of the algorithm.

Application to data integration and discovery

Integration of computational methods described above establishes a complete data integration and discovery framework for efficient data discovery, harmonization and integration of biobank data. Users can first discover suitable biobanks by searching for topics in BiobankUniverse. They can then upload their desired data schema to determine the harmonization potential of the candidate biobanks. Finally, users can harmonize the biobank data against the target data schema using SORTA to recode data to a standard code system and use MOLGENIS/connect to generate the data integration rules. The relationships between our tools and the common building blocks underlying them can be seen in **Figure 1**.

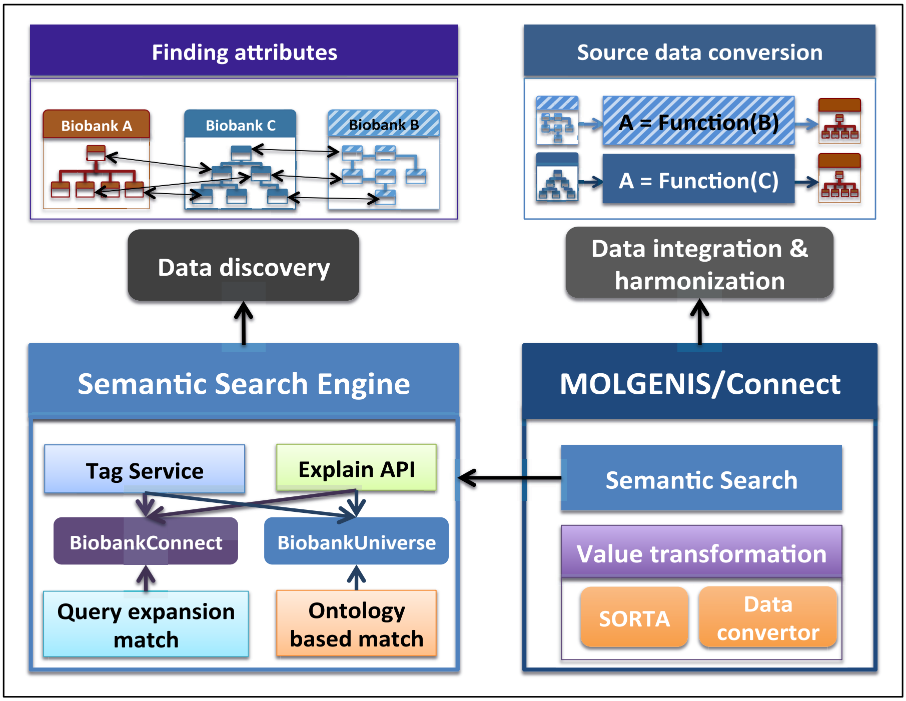


Figure 1 | **Overview of the data discovery and data harmonization framework.** The semantic search engine consists of four different components: TagService, ExplainAPI, QueryExpansionMatch and OntologyBasedMatch. BiobankUniverse and BiobankConnect share two common components: TagService and ExplainAPI. BiobankConnect uses the query-expansion-based matching algorithm while BiobankUniverse uses the ontology-based matching algorithm. BiobankUniverse is much more efficient than BiobankConnect at handling large biobanks and ontologies (e.g. UMLS). MOLGENIS/connect uses Semantic Search Engine to find the candidate attribute matches for data harmonization and integration. For converting source values to the standard target schema, SORTA and the MOLGENIS/connect built-in data converter can make suggestions for data transformation algorithms.

6.2 Evaluation of the methods

Speeding up data discovery

Discovering biobank data relevant to a given project is always the first step [84]. The traditional process for discovering datasets was typically that researchers would ask their professional friends (colleagues or collaborators) for the information and, usually, one of them (or possibly a friend of this friend) would happen to know of potentially useful datasets. This type of local and informal data discovery, however, is limited by an individual's professional network and networks' cumulative knowledge. With the development of search engines like Google, we can now easily access a lot more information, but the search can be also very hard because the webpages hosting the datasets don’t usually contain the proper metadata. The search for appropriate datasets can be further complicated if the data within different biobanks was generated using different terminologies (different code systems or ontologies) and structures (data models). In addition, some institutes still ‘follow’ non-standardized practices in designing questionnaires and data collection. The most difficult problem here are data that is collected as free text, e.g. in the reporting of diseases or physical activity. Although many efforts have been made to tackle data collection heterogeneities, such as establishing the standards, translating terminologies and matching data models, it can still take years of project participant time and the associated salary to build the IT infrastructures needed to utilize all of the existing efforts and enable the seamless data flow between data owners and researchers. Data discovery systems can be classified into two different types: centralized and dynamic data discovery systems (**Figure 2**).

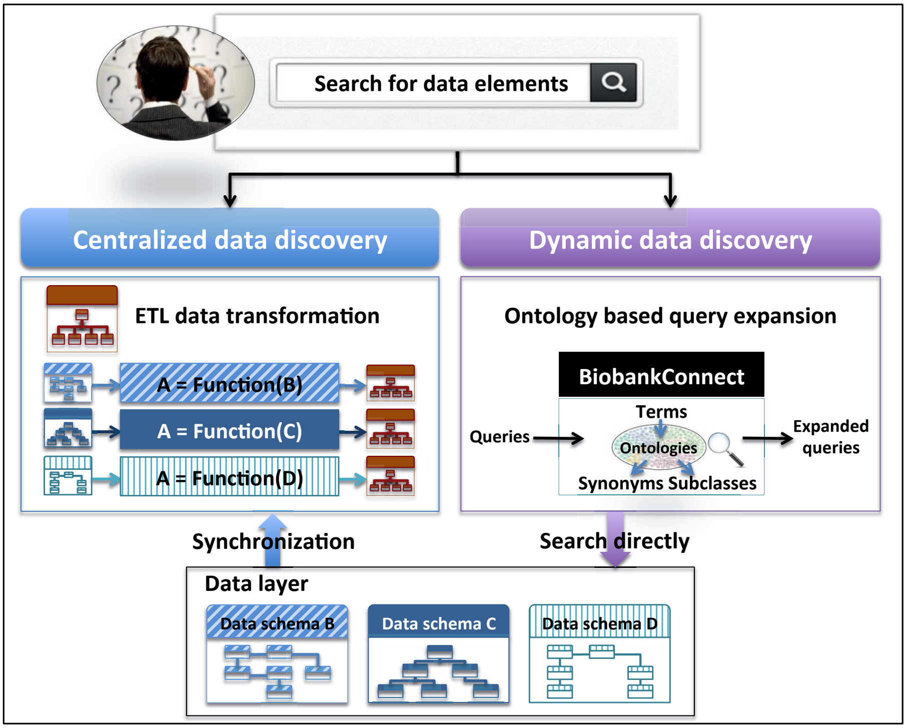


Figure 2 | **The two types of data discovery systems**. In Centralized data discovery, the source data have been projected onto the same data standard and the values have been transformed accordingly. In Dynamic data discovery, the source data are tagged with ontology terms. The user query dynamically gets turned into ontology terms, which are then used to search for the data elements with the relevant tagged ontology terms.

Differences between integration and search-based discovery

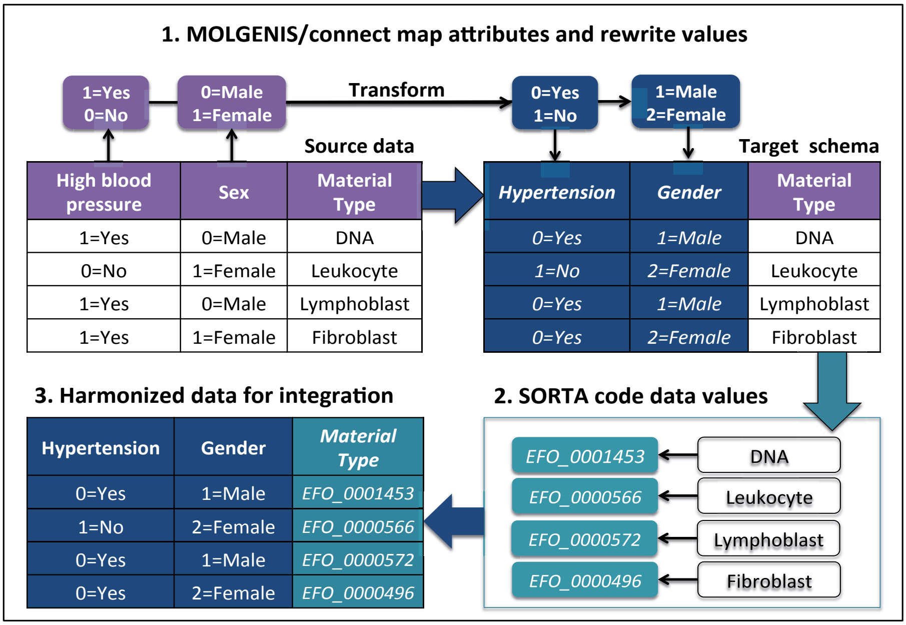
To reduce the workload of associated with fruitless searching, some consortia have put a lot of effort into setting up centralized data warehouses to serve as the entry point for researchers to locate relevant information. However, this approach requires data owners to agree on a standard data input model and far technicians to harmonize source datasets for data integration. Setting up such a system is essentially an ETL (extract, transform and load) data integration process [13]. The procedure is the same as that described in the previous section ‘Speeding up data integration’. However, data integration systems suffer from the limitation that deep data discovery with more fine-grained search queries cannot yet be achieved. Therefore we need a complementary dynamic system such as BiobankUniverse in which users can easily search for all source attributes based on custom defined queries, providing the flexibility to retrieve more relevant information. BiobankUniverse supports the two different types of discovery, which are described below.

Discovery by topic: Users can query topics in BiobankUniverse to find attributes of interest from all available datasets. The user queries are turned into a list of ontology terms, which are then used to search for relevant attributes using information from ontology terms. BiobankUniverse retrieves not only the relevant attributes but also the matching correspondences between them. By doing so, it provides users with an overall idea of which datasets overlap and on what topics.

Discovery by overall similarities: Users can upload their own data schemas to BiobankUniverse to quickly discover potential biobank source data for their analyses. The uploaded data schemas are automatically tagged with the UMLS ontology and then added to the universe. The system subsequently generates pairwise matches and semantic similarity scores between the data schema and all the existing biobanks. Based on the semantic scores or the number of generated matches, users are able to quickly identify which biobank data might be useful for their research question.

Speeding up data harmonization & integration

Using BiobankConnect and SORTA, a complete data integration task can be carried out efficiently. **Figure 3** shows a conceptual example of integrating one particular source dataset into the target central database involving two different types of harmonization. In the first harmonization step, MOLGENIS/connect is used to harmonize the differences at the metadata level between the target and the sources. Consequently, the source data values need to be adapted to the changes in the metadata and transformed according to the definition of the target, e.g. modifying the source column names, adjusting the categories and converting the data units. In the second harmonization step, SORTA is applied to harmonize string-type data values by matching them with the standard coding systems (ontologies) in order to achieve interoperability at the data level. As shown in this example, using both components can help us quickly set up a centralized data repository.


1. MOLGENIS/connect map attributes and rewrite values
1=Yes 0=No
0=Male 1=Female
Transform
0=Yes 1=No
1=Male 2=Female
Source data
Target schema
3. Harmonized data for integration
2. SORTA code data values

High blood pressure	Sex	Material Type
1=Yes	0=Male	DNA
0=No	1=Female	Leukocyte
1=Yes	0=Male	Lymphoblast
1=Yes	1=Female	Fibroblast

Hypertension	*Gender*	Material Type
0=Yes	*1=Male*	DNA
1=No	*2=Female*	Leukocyte
0=Yes	*1=Male*	Lymphoblast
0=Yes	*2=Female*	Fibroblast

EFO_0001453	DNA
EFO_0000566	Leukocyte
EFO_0000572	Lymphoblast
EFO_0000496	Fibroblast

Hypertension	Gender	*Material Type*
0=Yes	1=Male	*EFO_0001453*
1=No	2=Female	*EFO_0000566*
0=Yes	1=Male	*EFO_0000572*
0=Yes	2=Female	*EFO_0000496*

Figure 3 | **Setting up a data integration project using MOLGENIS/connect plus SORTA**.

6.3 Related developments and broader application

Data discovery and interoperability has garnered much interest over the past few years because of the needs of researchers for larger data sets to reach statistical significance and because of the desire of funding bodies to maximize reuse of existing data and knowledge the increase the returns on research funding (in contrast to investing in *de novo* data generation for each project). Enabling seamless data flow between different systems will ultimately enable reuse of scientific outputs and the discovery of knowledge. However, there is still work ahead of us before reaching this destination.

We believe the work described in this thesis can contribute to this ideal infrastructure for science. Interestingly, our work contributes perfectly to the recently evangelised principles of Findability, Accessibility, Interoperability and Reusability (FAIR), which are now widely accepted goals when describing data-intensive science. In addition, we have created a hybrid system that integrates the two most widely used techniques in the field, Semantic Integration (often used in context of 'Linked data') and use of Extract, Transform and Load (ETL) data integration methods.

Tools to retrospectively make data comply to FAIR principles

In 2014, a group of data scientists, funding bodies, publishers and other stakeholders held a workshop in Leiden to formulate the ideal principles for storing and sharing electronic data records in scientific discourse. These principals were ultimately summarized as findability, accessibility, interoperability and (re)usability (FAIR) [85]. The motivation for coming up with these principles is that good data management and data stewardship are essential to enable discovery and reuse of scientific knowledge and data as a basis for reproducible science. Ideally all data repositories should follow these FAIR guidelines in order to help users discover the 'right' datasets for their research. However, many questions remain about the details of how this can be achieved.

In this thesis we have developed computational methods that can retrospectively 'FAIR-ify' research data, making existing data adhere to FAIR principles. Our data integration suite (MOLGENIS/connect + SORTA) can harmonize data based on any given schema(s) therefore providing the ultimate flexibility to make the data compatible and interoperable, and with constantly evolving standards (for example as described in http://biosharing.org). Our data discovery systems (mostly semantic-search-based) make data findable using ontology-based semantic searching so that users can quickly discover the relevant information. In addition, all of our systems are built based on the MOLGENIS platform, in which data can be easily accessed via either a REST-API interface or, for those with permissions, downloaded from the standard data explorer, which is one of the accepted methods for interoperable data access. We plan to continue to develop the tools described in this thesis to become a new system called the MOLGENIS/fairifier that also implements emerging standard software interfaces for FAIR data exchange.

Semantic web and linked data

Many experts in semantics would argue that semantic web technology is 'the one and only solution' to all data integration problems in the biomedical domain. Semantic web technology allows us to capture the richness of the data, particularly for information that is buried in the documentation, e.g. high blood pressure is measured 10 times in the LifeLines biobank and an average value is taken as the final value. Bianchi *et al* [86] described a framework for combining multiple cohort studies using semantic integration. They created an ontology representation of the source data and a common data scheme and they mapped all the classes from the local ontology to the common terminologies such as SNOMED-CT and LOINC. However, what hinders researchers from using this approach is the fact that it requires them to properly ontologize the source data, a very time-consuming step. In addition, there are hundreds of bio-ontologies available and, while there are some standards about which ontologies should be used for certain domains, the choices of ontologies can be inconsistent among users. One user may annotate diseases using Human Disease Ontology (DO), while another uses the International Classification of Diseases (ICD).

To popularize the semantic web approach there are few things we have done to ease the technical burden to the users. Firstly, researchers need to be convinced of the benefits of transforming their data to the ontology representation, which we have demonstrated using ontology tagging throughout the system. Such recoding to ontology terms is the basis for participating in the linked data world. Secondly, the matches between ontologies need to be constantly updated and improved so that biobank data annotated using different ontologies can be easily exchanged. Thirdly, better ontology term annotators are needed to process massive amount of biobank attributes. For exampled, the official UMLS annotator MetaMap annotates ‘History of myocardial infarction’ with C0155668:Old myocardial Infarction [synonym:History Myocardial Infarction] and C1275835:History of Myocardial Infarction but fails to find the atomic ontology terms such as ‘History’ AND ‘Myocardial infarction’, which usually carry more information such as super/subclasses and synonyms than the perfectly matched ontology terms.

Semantic web technologies especially are especially beneficial for those data sources that have been properly annotated with ontology terms because BiobankUniverse and MOLGENIS/connect make use of the accurate ontology term annotations to find high quality attribute matches between biobanks for data discovery and integration. And, while our systems don’t directly support the communication with semantic web-based applications, we plan to publish the attribute matches generated by our systems in RDF format for semantic web researchers who prefer to solve the integration and discovery problems using the other approach. Our matches can then, for example, be used as a validation set to verify whether or not the accurately annotated ontology terms can conclude the same set of matches.

Traditional Extract, Transform and Load integration

Extract, Transform and Load (ETL) integration is the traditionally used approach for data integration and many tools have been built based on this procedure. OPAL and transMART [19,87] are two popular integration tools extensively using ETL in multiple projects. Their ETL procedures are quite similar. The data element matches are first identified between the target schema and source datasets, then a set of transformation algorithms are defined based on the matches, and these are used to pool data from multiple sources based on the same target schema. However, the harmonization work is done manually in both OPAL and transMART.

Several researchers using OPAL and tranSMART have proposed to adding the computational tools described in this thesis as a 'pre-processor' for the ETL procedure. This kind of data integration is a very flexible but complex process with a lot of exceptions and variations that make it difficult to automate completely. In order to speed it up, we broke the whole procedure down into small steps, automating part of the each process where possible, then connected these steps into a seamless workflow (MOLGENIS/connect). The result was greatly improved productivity of integration. Notably, we have two important components, the semantic search and the algorithm generator, which can work both together and independently. The semantic search in our systems can automatically provide the candidate matches for generating the algorithms and the algorithm generator can make use of those matches to generate transformation algorithms.

6.4 Suggestion for methodological enhancement

The computational methods used in this project are based on lexical matching, semantic matching and semantic query expansion. However, there are other computational methods available that might be used to improve data integration systems in the future:

Natural language processing

In BiobankUniverse, we developed a tagging service to automatically find ontology terms for given biobank attributes. The tagging algorithm is optimized to capture as many ontology terms as possible by producing a match when any of the synonyms of the ontology terms are found within the words of the attribute label. Our motivation was that we needed to not only find perfectly matched ontology terms but also combinations of partially matched terms.

However, while the matching criteria can make sure that the system doesn’t miss any important ontology terms it will also introduce unexpected ones. For example, the target attribute ‘Currently Follows a Cholesterol Lowering Diet’ is tagged with a group of ontology terms [‘Diet followed’ & ‘Cholesterol-lowering diet (finding)’ & ‘Cholesterol’]. When matching this target attribute in biobanks, the presence of ‘Cholesterol’ will inevitably lead to unwanted false positives. So, how can we automatically detect the important concepts in the label of the biobank attributes and only use those concepts to find the ontology terms?

We executed preliminary experiments that suggest Natural Language Processing (NLP) might help [83]. NLP automatically detects relationships between words in a sentence and decides which of the connected words can form a phrase or concept. In **Figure 4a**, ‘History’ and ‘Myocardial Infarction’ are identified as two separate concepts that should be matched to the ontology terms. In **Figure 4b**, ‘Cholesterol Lowering Diet’ is identified as the whole concept and hence ‘Cholesterol’ shouldn’t be tagged by ontology terms on its own.

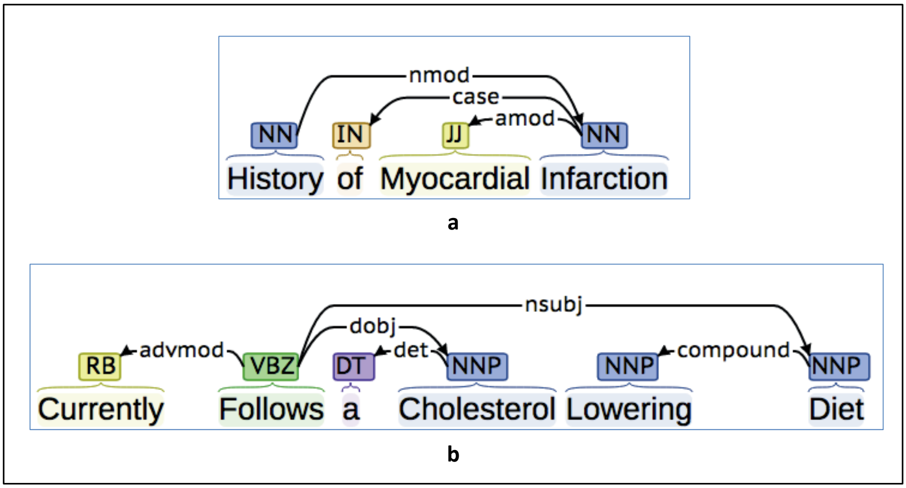

nmod
case
amod
NN
IN
JJ
NN
History of Myocardial Infarction
a
nsubj
dobj
advmod
det
compound
RB
VBZ
DT
NNP
NNP
NNP
Currently Follows a Cholesterol Lowering Diet
b

Figure 4 | **Analysis of sentences with the Stanford coreNLP tool.** In the sentence in **a**) **Myocardial** is an adjectival modifier (amod) for the succeeding noun **Infarction** (NN) therefore forming a concept/phrase. **History** (NN) is a noun on its own in the sentence and therefore detected as another concept. In the sentence in **b**) **Cholesterol** (NNP), **Lowering** (NNP) and **diet** (NNP) are three connected nouns forming a concept.

Similarly, the semantic search algorithm provides the matched words for both the target and the source attributes, which are important indicators that allow users to quickly decide whether the attribute pair is a good match. Since NLP tools can discover the concepts (phrases) from the attribute labels, we can let the system compare the matched words with those detected concepts and decide whether to keep or to remove such a candidate match. For example, the target attribute ‘Currently Follows a Cholesterol Lowering Diet’ is matched with the source attribute ‘Sodium restricted diet’ based on the word ‘diet’. In parallel, we analysed the two attribute labels using the Stanford coreNLP library [83] to find that the target concept is ‘cholesterol lowering diet’ and the source concept is ‘sodium restricted diet’. However, the matched word produced by the semantic search algorithm is ‘diet’, which is not matched to either of the concepts, and was therefore removed from the candidate list.

Machine learning

The common feature of all the tools developed within this thesis is that they all have semantic matching as the underlying functionality, ultimately producing a list of candidate matches for users to choose from. Finding one optimal cut-off value that yields the best precision and recall is a sound solution, but might not be discriminative enough because only one feature (i.e. the cut-off for similarity scores) is used for classification. After we published BiobankConnect, Ashish *et al* [88] demonstrated use of a machine learning based approach to find matching correspondences between target and source entities in the context of Alzheimer’s disease. We might have better discrimination if we would introduce similar machine learning methods that train the system to find dynamic cut-off values depending on multiple more subtle features.

The Ashish *et al* system, which is similar to BiobankConnect, produces a list of candidate data element matches for users to choose from. To achieve that, they synthesized a list of the features used to train a binary classifier with the Sequential Minimal Optimization algorithm to predict whether or not the candidate matches are relevant. Those features include 1) similarity measures such as Term Frequency and Inverse Document Frequency (TF-IDF) based and topic model based similarity scores; 2) metadata constraints, e.g. value ranges for numeric elements and cardinality for categorical elements (the number of possible values for an element); and 3) queries of whether the target and the source elements come from similar tables, e.g. the disease table or the demographics table. In addition, Ashish *et al* implemented an active learning mechanism that allows the system to be trained continuously as users produce more training data by selecting the correct matches.

Inspired by their work, we first conceptually discussed the potential features for the prediction model and then experimented them in a preliminary test run to decide on the final features, the 2results of which we will describe below. The complete list of features we used to train the system is shown in **Table 1**.

Table 1 | Features used for training the neural network model in R.

Feature	Description
2gramNameScore	A 2-gram similarity score calculated between the names of the data elements
vsmScore	An ontology term based Vector Space Model cosine similarity calculated between the labels of the data elements (described in **Chapter 5)**. The labels of the target and the source data element are partially replaced with ontology terms prior to the calculation, e.g. 'beans' ➔ 'vegetables'
vsmScoreRank	Rank produced based on the vsmScore among candidate matches for the same target data element.
2gramScore	Ontology term based 2-gram similarity score calculated between the labels of the data elements. The process is same as the vsmScore except that a 2-gram similarity is calculated instead.
2gramScoreRank	Rank produced based on the 2gramScore among candidate matches for the same target data element.
wordVectorScore	Using GloVe, an unsupervised learning algorithm, we obtained vector representations for all the words from all the labels of data elements. Based on this, we calculated the cosine similarity between target and source data elements.
wordVectorScoreRank	Rank produced based on wordVectorScore among candidate matches for the same target data element.
sourceMatchedWordIDF	Summed inverse document frequency of matched words among all source candidate matches for the same target data element.
sourceMatchedWordFrequency	Occurrence of matched words among all source candidate matches for the same target data element.

We included the most common features, such as different similarity scores and the ranks produced by them. We also included the number of matched words in the candidate matches and their corresponding inverse document frequencies. Moreover, we argue that sourceMatchedWordFrequency is another useful feature because, for the same target data element, matched word frequencies, calculated based on all candidate matches, may indicate the distribution of topics so the candidates generated based on matched words with low frequencies should be treated as less important. We did not include data type as a feature because the common data type constraints don't always work in biobank analyses. While a categorical data element may be matched to a decimal data element, in practice this could be rather more complicated. For instance, the decimal data element 'the number of years of education' is matched to the categorical data element 'Education', which has a list of education levels such as, 'primary school', 'high school' etc. Depending on the source country, the 'the number of year of education' can be deduced from the education the person has received so far.

We used the neuralnet package in R [89] to train the model based on the selected features with one hidden layer that consists of 10 units that predict whether or not the candidate matches are the relevant. We evaluated this model in four independent matching experiments for which we have the true matches: HOP-Lifelines, HOP-Mitchelstown, HOP-Prevend, and Finrisk2002-Finrisk2007. We conducted a scenario where users were randomly asked to curate 30% of all candidate matches as they would in BiobankUniverse. We then randomly split the curated matches into a training set (75% * 30% = 22.5%) and a validation set (25% * 30% = 7.5%). Because the model could be made to converge at the local minimums, we used the validation set to find the one solution that maximizes the f-measure. Based on the final model, we predicted the relevance for the rest of the candidate matches (70%) and calculated recall, precision and f-measure.

Table 2 shows our preliminary results. Interestingly, the performance in the Finrisk2002-Finrisk2007 matching experiment is much better than the others. This is because the two biobanks are much more similar than the others as they were developed within the same project. To elucidate the causes of the variability in performance, we plotted the distributions of the similarity scores between the relevant and irrelevant matches across the four matching experiments (**Figure 5)**. Within Finrisk, the distributions of the vsmScore are well distinguished between the positive and the negative cases, while in the other experiments the scores are mixed and can't be separated. These mixed results suggest more research is needed before machine learning can be used to improve the performance of the methods described in this thesis.

Table 2 | Evaluation of the neural network model in four independent matching experiments. The neural network model is trained based on the list of features listed in **Table 1** and has outputs 1 or 0 (relevant or irrelevant) for each candidate match. The predictions are then compared with observations to compute Recall and Precision to indicate the performance.

Matching experiment	Recall	Precision	F-measure
HOP-LifeLines	0.584	0.440	0.502
HOP -Mitchelstown	0.450	0.636	0.527
HOP -Prevend	0.216	0.222	0.219
Finrisk2002-Finrisk2007	0.768	0.928	0.841

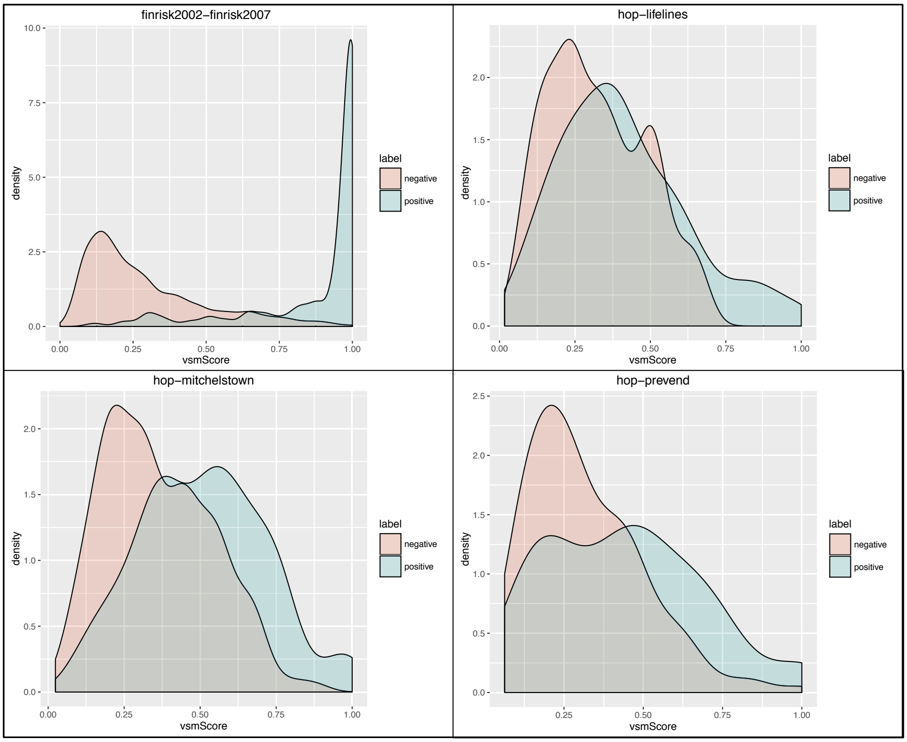


Figure 5 | **vsmScore distributions between true matches and false matches across four independent matching experiments**. The distributions of the other two similarity scores (2gramScore and wordVectorScore) show similar trends and are therefore not shown.

6.5 Conclusion

In this thesis, we have developed computational methods to overcome barriers to data discovery, harmonization and integration. We have demonstrated that implementation of these methods in user friendly tools can free researchers from most of the manual effort and time burden of data transformation or data discovery and can allow them to focus on answering research questions. We hope our work will further enable 'FAIR' data reuse to improve scientific efficiency and reproducibility in this exciting 'data deluge' era of the biomedical sciences.

Supplementary Information

Supplementary Table S1

Precision and Recall calculated based on n-gram similarity cutoffs from 80% to 100%

	Before curation		After curation	
n-gram cutoff	Recall	Precision	Recall	Precision
80%	0.22	0.99	0.78	0.99
81%	0.22	0.99	0.75	0.99
82%	0.22	1.00	0.73	0.99
83%	0.21	1.00	0.73	0.99
84%	0.20	1.00	0.71	1.00
85%	0.20	1.00	0.70	1.00
86%	0.19	1.00	0.68	1.00
87%	0.18	1.00	0.68	1.00
88%	0.18	1.00	0.67	1.00
89%	0.18	1.00	0.66	1.00
90%	0.17	1.00	0.65	1.00
91%	0.17	1.00	0.64	1.00
92%	0.17	1.00	0.63	1.00
93%	0.16	1.00	0.60	1.00
94%	0.16	1.00	0.58	1.00
95%	0.15	1.00	0.57	1.00
96%	0.14	1.00	0.56	1.00
97%	0.14	1.00	0.56	1.00
98%	0.14	1.00	0.56	1.00
99%	0.14	1.00	0.56	1.00
100%	0.14	1.00	0.56	1.00

Supplementary Table S2

Precision and Recall based on n-gram similarity from 88% to 100%

N-gram cutoff	Recall	Precision
88%	0.33	0.98
89%	0.33	1.00
90%	0.32	1.00
91%	0.30	1.00
92%	0.29	1.00
93%	0.29	1.00
94%	0.29	1.00
95%	0.28	1.00
96%	0.28	1.00
97%	0.28	1.00
98%	0.28	1.00
99%	0.28	1.00
100%	0.28	1.00

Supplementary Table S3

Comparison of performances for SORTA, BioPortal Annotator and ZOOMA in recreating existing ontology matches. The evaluation was based on three pairs of existing ontology matches: HPO/DO, HPO/NCIT and HPO/OMIM. The table shows the recall/precision per rank in SORTA, BioPortal Annotator and ZOOMA.

				Matching task for HPO-DO (700 matches)					
	SORTA			BioPortal Annotator			ZOOMA		
Rank cutoff	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
1	0.999	0.999	0.999	0.964	0.974	0.969	0.979	0.994	0.986
2	1.000	0.500	0.999	0.964	0.756	0.847	0.983	0.984	0.984
3	1.000	0.330	0.500	0.964	0.731	0.832	0.984	0.983	0.984
				Matching task for HPO-NCIT (1148 matches)					
	SORTA			BioPortal Annotator			ZOOMA		
Rank cutoff	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
1	0.997	0.997	0.997	0.979	0.988	0.984	0.987	0.996	0.992
2	1.000	0.500	0.667	0.979	0.792	0.876	0.988	0.967	0.977
3	1.000	0.333	0.500	0.979	0.770	0.862	0.989	0.963	0.976
				Matching task for HPO-OMIM (3631 matches)					
	SORTA			BioPortal Annotator			ZOOMA		
Rank cutoff	Recall	Precision	F-measure	Recall	Precision	F-measure	Recall	Precision	F-measure
1	0.996	0.996	0.996	0.670	0.974	0.794	0.976	0.993	0.984
2	1.000	0.500	0.667	0.670	0.761	0.713	0.980	0.987	0.983
3	1.000	0.333	0.500	0.670	0.732	0.700	0.980	0.986	0.983

HPO: Human Phenotype Ontology; DO: Disease Ontology; NCIT: National Cancer Institute Thesaurus; OMIM: Online Mendelian Inheritance in Man

Supplementary Table S4

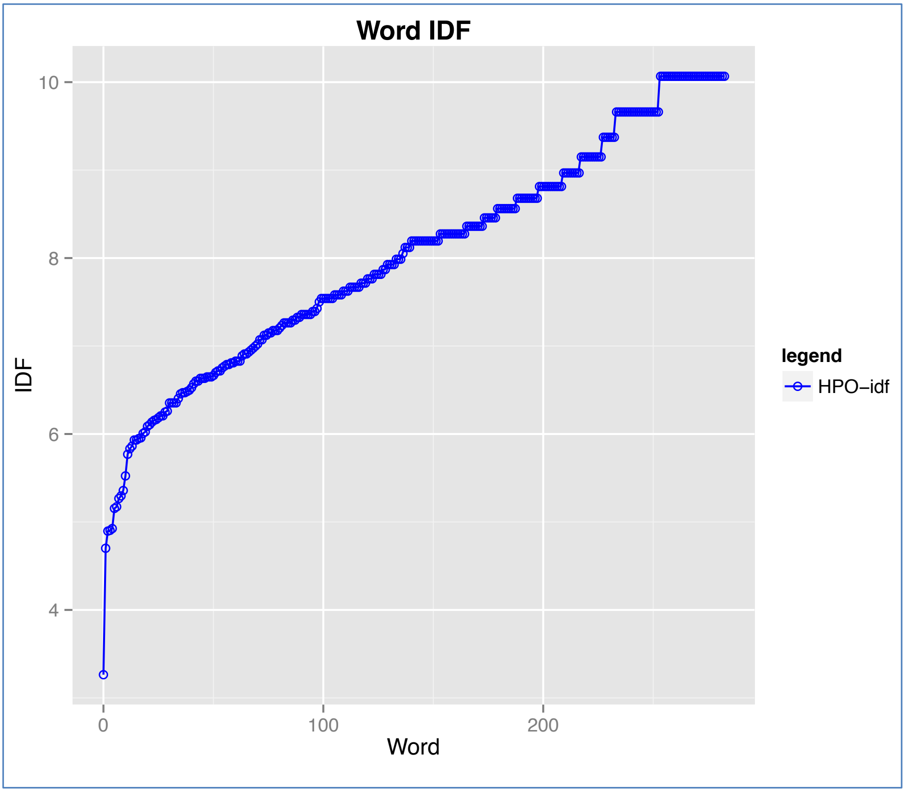
Evaluation of performance for SORTA at different percentage cutoff values

Cut-off percentage	HPO-DO (700 matches)		HPO-NCIT (1148 matches)		HPO-OMIM (3631 matches)	
	Recall	Precision	Recall	Precision	Recall	Precision
100%	0.990	1.000	0.993	1.000	0.995	1.000
90%	0.996	1.000	0.999	1.000	0.996	1.000
80%	0.999	0.999	1.000	1.000	1.000	1.000

HPO Human Phenotype Ontology; DO: Disease Ontology; NCIT: National Cancer Institute Thesaurus; OMIM: Online Mendelian Inheritance in Man

Supplementary Figure S5

The inverse document frequency (IDF) for input query words. The IDF is first calculated for all the words available from Human Phenotype Ontology (HPO) to create the IDF library, then all of the words from the input query are checked against this library to create the plot.



Supplementary Table S6

Matching amount categories. Example of complex matches between target and source categories and the corresponding quantified amount that describes the frequency of ‘potato consumption’ for the target attribute and the source attribute. First, the categories are converted to quantifiable amounts based on the key information (time unit and frequency) extracted from the description using regular expressions. Then, the source categories are matched to the target categories by determining the closest target amounts for source amounts.

Target **Current Consumption Frequency of Cooked Vegetables**		**Source** **Cooked vegetables**	
Categories	**Amounts**	**Categories**	**Amounts**
Never + less than once a week	Unit: week Frequency: 0-1	Less often than once a month or not at all	Unit: month Frequency: 0-1
		1-3 times a month	Unit: month Frequency: 1-3
About once a week	Unit: week Frequency: 1	Once a week	Unit: week Frequency: 1
Several times a week	Unit: week Frequency: 2-7	2-4 times a week	Unit: week Frequency: 2-4
		5-6 times a week	Unit: week Frequency: 5-6
Almost daily + daily	Unit: day Frequency: 1	Once a day	Unit: day Frequency: 1
		2-3 times a day	Unit: day Frequency: 2-3
		More than 4 times a day	Unit: day Frequency: 4

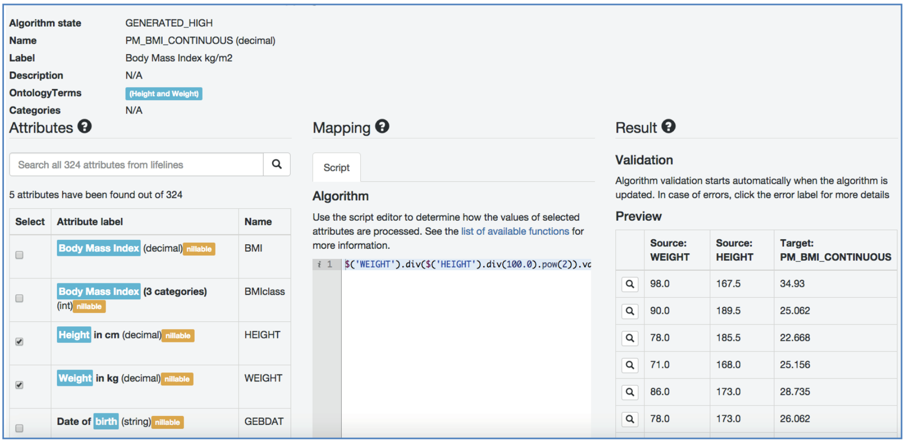
Supplementary Table S7

Matching complex categories. Pre-defined rules for matching categories and example applications.

Rule	Description	Example	
Rule1	Category label containing word '**No**' can be matched to the category that contains '**Never**'	No	Never had stroke
Rule2	Category label containing word '**Yes**' can be matched to the category that contains '**Ever**'	Yes	Ever had stroke
Rule3	Category label containing word '**Yes**' can be matched to the category that contains '**Has**'	Yes	Has had stroke
Rule4	Category label containing word '**Unknown**' can be matched to the category that contains '**Missing**'	Unknown	Missing
Rule5	Category label containing word '**Not know**' can be matched to the category that contains '**Missing**'	I do not know	Missing

Supplementary Figure S8

Overview of the algorithm editor.



Algorithm state	GENERATED_HIGH
Name	PM_BMI_CONTINUOUS (decimal)
Label	Body Mass Index kg/m2
Description	N/A
OntologyTerms	(Height and Weight)
Categories	N/A

Attributes

Search all 324 attributes from lifelines

5 attributes have been found out of 324

Select	Attribute label	Name
□	Body Mass Index (decimal) nillable	BMI
□	Body Mass Index **(3 categories)** (int) nillable	BMIclass
☑	Height **in cm** (decimal) nillable	HEIGHT
☑	Weight **in kg** (decimal) nillable	WEIGHT
□	**Date of** birth (string) nillable	GEBDAT

Mapping

Script

Algorithm

Use the script editor to determine how the values of selected attributes are processed. See the list of available functions for more information.

```
$('WEIGHT').div($('HEIGHT').div(100.0).pow(2)).vc
```

Result

Validation

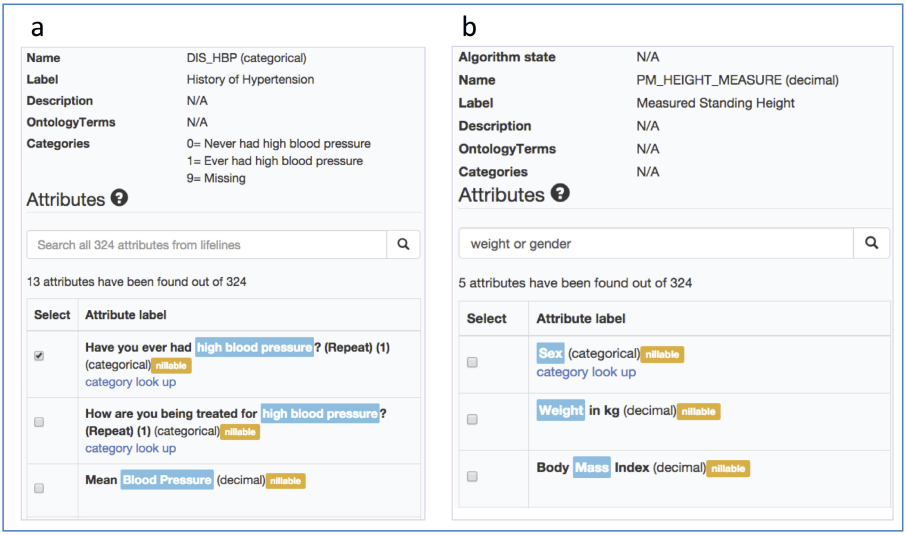
Algorithm validation starts automatically when the algorithm is updated. In case of errors, click the error label for more details

Preview

	Source: WEIGHT	Source: HEIGHT	Target: PM_BMI_CONTINUOUS
🔍	98.0	167.5	34.93
🔍	90.0	189.5	25.062
🔍	78.0	185.5	22.668
🔍	71.0	168.0	25.156
🔍	86.0	173.0	28.735
🔍	78.0	173.0	26.062

Supplementary Figure S9

Source attribute selector view (a) The target attribute was automatically annotated with the ontology term ‘Hypertension’. All the synonyms and subclasses of ‘Hypertension’ were used for query expansion. Based on these, Lucene retrieved 13 relevant attributes from the LifeLines database source. The words Lucene used to match are highlighted. (b) The semantic search box allows the user to optionally search all source attributes. When a user types in a term, it will also be automatically annotated with ontology terms to enable query expansion as described above. The user-defined query terms have the highest priority and only these are used in semantic search. The attribute label, description and existing ontology term annotations will not be used for query expansion if there are user-defined queries.



a

Name	DIS_HBP (categorical)
Label	History of Hypertension
Description	N/A
OntologyTerms	N/A
Categories	0= Never had high blood pressure 1= Ever had high blood pressure 9= Missing

Attributes

Search all 324 attributes from lifelines

13 attributes have been found out of 324

Select	Attribute label
☑	**Have you ever had high blood pressure? (Repeat) (1)** (categorical) nillable category look up
□	**How are you being treated for high blood pressure? (Repeat) (1)** (categorical) nillable category look up
□	**Mean Blood Pressure** (decimal) nillable

b

Algorithm state	N/A
Name	PM_HEIGHT_MEASURE (decimal)
Label	Measured Standing Height
Description	N/A
OntologyTerms	N/A
Categories	N/A

Attributes

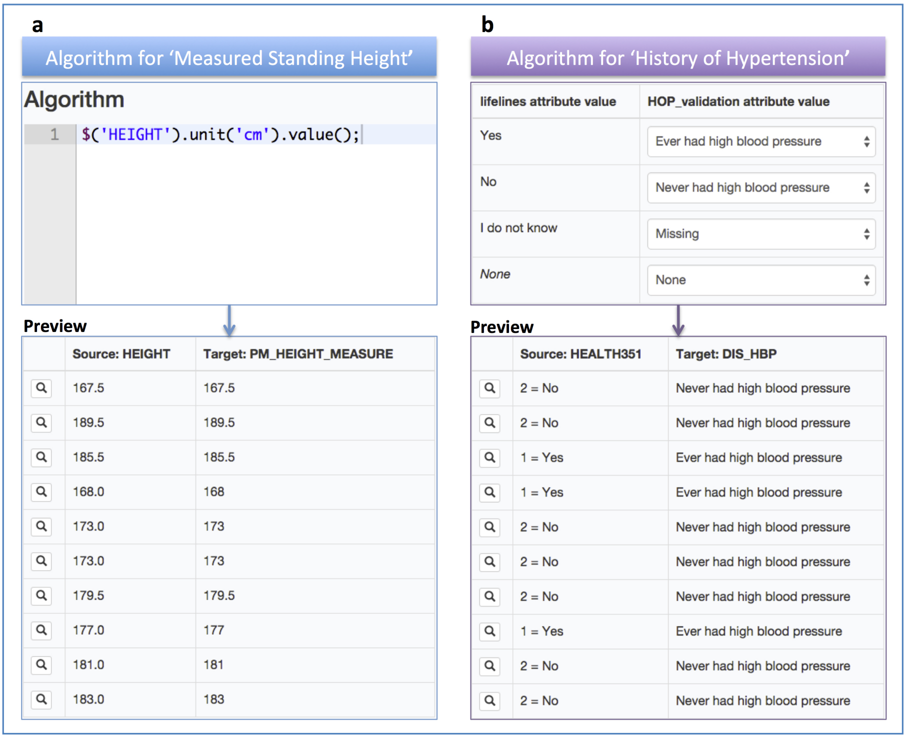
weight or gender

5 attributes have been found out of 324

Select	Attribute label
□	**Sex** (categorical) nillable category look up
□	**Weight in kg** (decimal) nillable
□	**Body Mass Index** (decimal) nillable

Supplementary Figure S10

Transformation algorithm editor (a) The auto-generated algorithm for the target attribute ‘Measured Standing Height’ from source attribute ‘Height at physical examination (m)’. The mention of the unit **m** in the source attribute label is automatically detected and a unit (‘cm’) convertor added to algorithm. A preview of the algorithm conversion results is provided for the user to check. (b) Since the target attribute and the source attribute are both categorical, a category-matching editor is provided for the user to easily match categories using a user interface.



Algorithm for ‘Measured Standing Height’

Algorithm

```
$('HEIGHT').unit('cm').value();
```

Preview

	Source: HEIGHT	Target: PM_HEIGHT_MEASURE
🔍	167.5	167.5
🔍	189.5	189.5
🔍	185.5	185.5
🔍	168.0	168
🔍	173.0	173
🔍	173.0	173
🔍	179.5	179.5
🔍	177.0	177
🔍	181.0	181
🔍	183.0	183

Algorithm for ‘History of Hypertension’

lifelines attribute value	HOP_validation attribute value
Yes	Ever had high blood pressure
No	Never had high blood pressure
I do not know	Missing
None	None

Preview

	Source: HEALTH351	Target: DIS_HBP
🔍	2 = No	Never had high blood pressure
🔍	2 = No	Never had high blood pressure
🔍	1 = Yes	Ever had high blood pressure
🔍	1 = Yes	Ever had high blood pressure
🔍	2 = No	Never had high blood pressure
🔍	2 = No	Never had high blood pressure
🔍	2 = No	Never had high blood pressure
🔍	1 = Yes	Ever had high blood pressure
🔍	2 = No	Never had high blood pressure
🔍	2 = No	Never had high blood pressure

Supplementary Table S11

Summary of the evaluations of the semantic search and the algorithm generator. For the algorithm generator, ‘Good’ means that the algorithms generated are either the same or equivalent to the manually created algorithms; ‘Partially-good’ means that the algorithms generated are very similar to the manually created algorithms, and can be easily fixed; ‘Bad’ means the algorithms generated are very far from the manually created algorithms. For the semantic search, ‘Good’ means that the attributes in the manually created algorithms are found within the top 20 suggested data elements; ‘Bad’ means that the attributes in the manually created algorithms are not found with the top 20 suggested data elements.

Topic	label	Prevend		LifeLines		Mitchelstown	
		Algorithm generator	Semantic search	Algorithm generator	Semantic search	Algorithm generator	Semantic search
Diet	Currently Follows a Cholesterol Lowering Diet	N/A	N/A	Partially-good	Perfect	Good	Perfect
	Currently Follows a Diabetic Diet	N/A	N/A	Partially-good	Perfect	Good	Perfect
	Currently Follows a Low Salt Diet	N/A	N/A	Partially-good	Perfect	N/A	N/A
	Currently Follows a Non-Gluten Diet	N/A	N/A	N/A	N/A	Good	Perfect
	Currently Follows a Vegetarian Diet	N/A	N/A	Bad	Good	Good	Good
	Currently Follows a Weight Loss Diet	N/A	N/A	Partially-good	Good	Good	Perfect
	Type of Vegetarian Diet	N/A	N/A	N/A	N/A	N/A	N/A
Disease	History of Diabetes	Good	Perfect	Good	Good	Good	Good
	History of Hypertension	Good	Perfect	Good	Perfect	Good	Perfect
	History of Myocardial Infarction	Good	Perfect	Good	Perfect	Good	Perfect
	History of Stroke	Good	Perfect	Good	Perfect	Good	Perfect
	Type of Diabetes	Partially-good	Good	Partially-good	Good	N/A	N/A
Drink	Current Quantity of Beer Consumed	N/A	N/A	N/A	N/A	Bad	Perfect
	Current Quantity of Spirits/Liquor Consumed	N/A	N/A	Bad	Good	Bad	Bad
	Current Quantity of Total Alcohol Consumed in Beer, Wine and Spirits per week	N/A	N/A	N/A	N/A	N/A	N/A
	Current Quantity of Wine Consumed	N/A	N/A	Bad	Good	Bad	Perfect
	Current Use of Alcohol	Partially-good	Perfect	Partially-good	Good	Partially-good	Perfect
	Level of current alcohol consumption	N/A	N/A	N/A	N/A	N/A	N/A
Education	Highest Level of Education	Bad	Good	Bad	Good	Partially-good	Perfect
	Highest Level of Education	N/A	N/A	N/A	N/A	N/A	N/A

Topic	label	Prevend		Lifelines		Mitchelstown	
		Algorithm generator	Semantic search	Algorithm generator	Semantic search	Algorithm generator	Semantic search
Education	Number of Years of Education	Bad	Perfect	N/A	N/A	Partially-good	Perfect
	Some Elements of Post-Secondary Non-Tertiary Education Completed	Partially-good	Perfect	Partially-good	Good	Partially-good	Perfect
	Some Elements of Tertiary Education Completed	Partially-good	Perfect	Partially-good	Good	Partially-good	Perfect
	Some Primary Education Completed	Partially-good	Perfect	Partially-good	Good	Partially-good	Perfect
	Some Secondary Education Completed	Partially-good	Perfect	Partially-good	Good	Partially-good	Perfect
Food	Current Consumption Frequency of Bakery Products	N/A	N/A	N/A	N/A	Good	Bad
	Current Consumption Frequency of Bread and Rolls	N/A	N/A	Good	Bad	Good	Bad
	Current Consumption Frequency of Breakfast Cereals	N/A	N/A	Good	Good	Good	Bad
	Current Consumption Frequency of Cheese	N/A	N/A	Good	Good	Good	Good
	Current Consumption Frequency of Chocolate	N/A	N/A	Good	Bad	Good	Bad
	Current Consumption Frequency of Chocolates/Sweets	N/A	N/A	Good	Bad	Good	Bad
	Current Consumption Frequency of Cooked Vegetables	N/A	N/A	Good	Good	N/A	N/A
	Current Consumption Frequency of Eggs	N/A	N/A	Good	Perfect	Good	Perfect
	Current Consumption Frequency of Fish	N/A	N/A	Good	Perfect	Good	Good

Topic	label	Prevend		Lifelines		Mitchelstown	
		Algorithm generator	Semantic search	Algorithm generator	Topic	label	Algorithm generator
Food	Current Consumption Frequency of Fruits	N/A	N/A	Good	Perfect	Good	Good
	Current Consumption Frequency of Meat and Meat Products	N/A	N/A	Good	Bad	Good	Bad
	Current Consumption Frequency of Milk	N/A	N/A	Good	Good	Bad	Bad
	Current Consumption Frequency of Nuts	N/A	N/A	Good	Good	Good	Perfect
	Current Consumption Frequency of Potatoes	N/A	N/A	Good	Good	Good	Good
	Current Consumption Frequency of Poultry and Poultry Products	N/A	N/A	N/A	N/A	Good	Bad
	Current Consumption Frequency of Raw Vegetables	N/A	N/A	Good	Good	N/A	N/A
	Current Consumption Frequency of Rice and Pasta	N/A	N/A	Good	Good	Good	Bad
	Current Consumption Frequency of Salted Snacks	N/A	N/A	Good	Bad	Good	Bad
	Current Consumption Frequency of Soft Drinks	N/A	N/A	Good	Good	Good	Bad
	Current Consumption Frequency of Sugar Products Excluding Chocolate	N/A	N/A	Good	Bad	Bad	Bad
	Current Consumption of Milk Products	N/A	N/A	Good	Bad	Bad	Bad
	Current Consumption Quantity of Coffee	N/A	N/A	Partially-good	Bad	Good	Good
	Current Consumption Quantity of Tea	N/A	N/A	Partially-good	Bad	Good	Perfect

Topic	label	Prevend		Lifelines		Mitchelstown	
		Algorithm generator	Semantic search	Algorithm generator	Semantic search	Algorithm generator	Semantic search
General	Age in Years	Bad	Bad	Bad	Bad	Good	Perfect
	Birth Year	Partially-good	Perfect	Partially-good	Perfect	Partially-good	Good
	Country of Birth	N/A	N/A	Partially-good	Good	N/A	N/A
	Current Country of Residence	Good	Perfect	N/A	N/A	N/A	N/A
	Current Region of Residence	N/A	N/A	N/A	N/A	N/A	N/A
	Gender	Good	Perfect	Good	Perfect	Good	Perfect
	Living with Partner	Bad	Bad	N/A	N/A	Partially-good	Bad
	Marital Status	N/A	N/A	N/A	N/A	Partially-good	Perfect
	Net Household Income	N/A	N/A	N/A	N/A	N/A	N/A
	Number of Live Births Mothered	N/A	N/A	N/A	N/A	Bad	Bad
	Number of People in the Household	N/A	N/A	N/A	N/A	Partially-good	Perfect
	Year of Interview	Partially-good	Bad	Partially-good	Bad	N/A	N/A
Job	Current Job Title (ISCO 88)	N/A	N/A	N/A	N/A	Partially-good	Bad
	Employment Status	Partially-good	Perfect	Partially-good	Bad	Partially-good	Bad
	Number of Working Hours	N/A	N/A	Partially-good	Perfect	N/A	N/A
	Retirement Status	Partially-good	Bad	Partially-good	Bad	N/A	N/A
	Student Status	N/A	N/A	Partially-good	Bad	N/A	N/A

Topic	label	Prevend		Lifelines		Mitchelstown	
		Algorithm generator	Semantic search	Algorithm generator	Semantic search	Algorithm generator	Semantic search
Measurement	Body Mass Index kg/m^2	Good	Perfect	Good	Perfect	Good	Perfect
	Creatinin	Bad	Bad	Good	Perfect	Good	Perfect
	Fasting Glucose	Bad	Bad	Good	Perfect	Partially-good	Perfect
	HDL Cholesterol	Good	Perfect	Good	Perfect	Good	Bad
	Hip Circumference	Good	Perfect	Good	Perfect	Bad	Good
	Inflammation Marker (hsCRP)	Good	Perfect	Good	Perfect	Partially-good	Perfect
	LDL Cholesterol (Friedewald Equation)	N/A	N/A	N/A	N/A	N/A	N/A
	Measured Diastolic Blood Pressure	Good	Perfect	Partially-good	Perfect	Bad	Good
	Measured Standing Height meter	N/A	N/A	Good	Perfect	Partially-good	Perfect
	Measured Systolic Blood Pressure	Good	Perfect	Partially-good	Perfect	Bad	Good
	Measured Weight kilogram	Good	Perfect	Good	Perfect	Good	Perfect
	Microalbuminuria	Bad	Bad	Bad	Bad	Partially-good	Bad
	Non-Fasting Glucose	Bad	Bad	N/A	N/A	N/A	N/A
	Total Serum Cholesterol	Good	Perfect	Good	Perfect	Partially-good	Perfect
	Triglycerides	Good	Perfect	Good	Perfect	Good	Perfect
	Waist Circumference	Good	Perfect	Good	Perfect	Bad	Good

Topic	label	Prevend		Lifelines		Mitchelstown	
		Algorithm generator	Semantic search	Algorithm generator	Topic	label	Algorithm generator
Medication	Current Use of Antihypertensive Medication	Partially-good	Good	Bad	Good	Partially-good	Perfect
	Current Use of Blood Glucose Lowering Medication	Partially-good	Perfect	Bad	Bad	Bad	Bad
	Current Use of Lipid Lowering Medication	Partially-good	Perfect	Bad	Bad	Bad	Bad
	Current Use of Lipid Lowering Medications Fibrates and/or Nicotinic Acid Derivatives	N/A	N/A	Bad	Good	Bad	Good
Smoking	Current Cigar Smoker	N/A	N/A	Bad	Bad	Bad	Good
	Current Cigarette Smoker	N/A	N/A	Bad	Bad	Bad	Good
	Current Pipe Smoker	N/A	N/A	Bad	Bad	Bad	Good
	Current Quantity of Cigarettes Smoked	Bad	Good	Bad	Bad	N/A	N/A
	Current Tobacco Smoker	Partially-good	Perfect	N/A	N/A	Partially-good	Good
	Ever Smoked Cigarettes	N/A	N/A	N/A	N/A	Bad	Good
	Ever Smoked Tobacco	Good	Good	N/A	N/A	Partially-good	Good
	Smoking status	N/A	N/A	N/A	N/A	Good	Perfect

Supplementary Table S12

List of semantic types labelled ‘unimportant’ and skipped in query expansion.

ID	Semantic Type	Semantic Type Group
T169	Functional Concept	Concepts & Ideas
T185	Classification	Concepts & Ideas
T081	Quantitative Concept	Concepts & Ideas
T079	Temporal Concept	Concepts & Ideas
T080	Qualitative Concept	Concepts & Ideas
T170	Intellectual Product	Concepts & Ideas
T078	Idea or Concept	Concepts & Ideas
T082	Spatial Concept	Concepts & Ideas
T070	Natural Phenomenon or Process	Phenomena
T204	Eukaryote	Living Beings
T045	Genetic Function	Physiology
T028	Gene or Genome	Genes & Molecular Sequences

Supplementary Table S13

Overall performance comparison using different levels of the ontology while expanding queries towards the parent direction. We calculated precision/recall/f-measure for six levels ranging from root level to level 6. For readability purposes, we only show the most interesting results from level 3 to level 6. For the full set of values see **Supplementary Material level_comparision.xls**.

Rank	Level3			Level4			Level5			Level6		
	R	P	F	R	P	F	R	P	F	R	P	F
1	0.25	0.52	0.34	0.25	0.53	0.34	0.25	0.66	0.36	0.27	0.61	0.37
2	0.39	0.43	0.41	0.39	0.45	0.42	0.38	0.55	0.45	0.41	0.50	0.45
3	0.46	0.37	0.41	0.47	0.39	0.42	0.46	0.47	0.47	0.48	0.43	0.46
4	0.53	0.33	0.41	0.53	0.35	0.42	0.52	0.44	0.47	0.54	0.40	0.46
5	0.57	0.31	0.40	0.58	0.33	0.42	0.58	0.42	0.48	0.58	0.37	0.45
6	0.61	0.29	0.39	0.61	0.31	0.41	0.60	0.39	0.47	0.61	0.34	0.44
7	0.63	0.27	0.38	0.64	0.29	0.40	0.64	0.37	0.47	0.63	0.32	0.42
8	0.66	0.26	0.37	0.67	0.28	0.39	0.67	0.36	0.47	0.65	0.31	0.42
9	0.68	0.25	0.36	0.68	0.27	0.38	0.69	0.35	0.46	0.67	0.30	0.41
10	0.70	0.24	0.35	0.70	0.26	0.38	0.71	0.34	0.46	0.69	0.29	0.41
11	0.71	0.23	0.34	0.71	0.25	0.37	0.72	0.33	0.45	0.70	0.28	0.40
12	0.73	0.22	0.34	0.73	0.24	0.37	0.74	0.32	0.45	0.71	0.28	0.40
13	0.74	0.22	0.34	0.74	0.24	0.36	0.75	0.32	0.45	0.72	0.27	0.40
14	0.76	0.21	0.33	0.76	0.24	0.36	0.77	0.32	0.45	0.74	0.27	0.40
15	0.77	0.21	0.33	0.77	0.24	0.36	0.78	0.31	0.45	0.75	0.27	0.40
16	0.78	0.21	0.33	0.78	0.23	0.36	0.79	0.31	0.44	0.75	0.27	0.39
17	0.79	0.20	0.32	0.79	0.23	0.36	0.80	0.31	0.44	0.76	0.26	0.39
18	0.79	0.20	0.32	0.80	0.23	0.35	0.80	0.30	0.44	0.77	0.26	0.39
19	0.80	0.20	0.32	0.80	0.23	0.35	0.81	0.30	0.44	0.77	0.26	0.39
20	0.81	0.20	0.32	0.81	0.22	0.35	0.82	0.30	0.44	0.77	0.26	0.39
50	0.84	0.16	0.27	0.84	0.20	0.32	0.85	0.26	0.40	0.80	0.22	0.35

P, precision; R, recall; F, f-measure

Supplementary Example S14

Attribute ‘Number of years of education’ is tagged with taggroup(**Number** && **Year** && **Education**) with 100% similarity score. However, there might be multiple ontology terms that are matched with same similarity scores and same words. The ontology terms matched with the same words are put into a temporary taggroup. Once we have collected members for all the temporary taggroups, all possible combinations of group members are generated from all the temporary groups. The possible taggroups that are generated from group A (A1;A2) and group B(B1;B2) are taggroup(A1 && B1), taggroup(A1 && B2), taggroup(A2 && B1), taggroup(A2 && B2). For example, ‘History of Hypertension’ is matched with three ontology terms, Hypertension (50%), history (30%), medical history [synonym:history] (30%). Two tag groups are generated from this list, taggroup(Hypertension && history) and taggroup(Hypertension && medical history).

Supplementary Example S15

In BiobankConnect, for example, a query on 'beer intake' in one biobank would not find the attribute ‘alcohol intake’ from another biobank, while the reverse query starting with the more general 'alcohol intake' would work. In BiobankUniverse, Sibling will get added to the expanded query for the ontology term Brother because of the existing relationship ‘Brother is a subClassOf Sibling’. More formally, an attribute with tag group (A && B), with A, B being ontology terms, now has an expanded query of ((Asub | Apar | Asyn) && (Bsub | Bpar | Bsyn)), with sub=subclasses, par=parent classes and syn=synonyms of ontology terms A,B.

Supplementary Table S16

Semantic types designated ‘unimportant’ and skipped during query expansion.

ID	Semantic Type	Semantic Type Group
T169	Functional Concept	Concepts & Ideas
T185	Classification	Concepts & Ideas
T081	Quantitative Concept	Concepts & Ideas
T079	Temporal Concept	Concepts & Ideas
T080	Qualitative Concept	Concepts & Ideas
T170	Intellectual Product	Concepts & Ideas
T078	Idea or Concept	Concepts & Ideas
T082	Spatial Concept	Concepts & Ideas
T070	Natural Phenomenon or Process	Phenomena
T204	Eukaryote	Living Beings
T045	Genetic Function	Physiology
T028	Gene or Genome	Genes & Molecular Sequences

Supplementary Table S17

Overall performance comparison using different levels of the ontology while expanding queries towards the parent direction. We calculated precision, recall and f-measure for six levels ranging from root level to level 6. We show only the relevant results: those from level 3 to level 6 for readability. See **Supplementary Material level_comparision.xls** for the full dataset.

Rank	Level 3			Level 4			Level 5			Level 6		
	R	P	F	R	P	F	R	P	F	R	P	F
1	0.25	0.52	0.34	0.25	0.53	0.34	0.25	0.66	0.36	0.27	0.61	0.37
2	0.39	0.43	0.41	0.39	0.45	0.42	0.38	0.55	0.45	0.41	0.50	0.45
3	0.46	0.37	0.41	0.47	0.39	0.42	0.46	0.47	0.47	0.48	0.43	0.46
4	0.53	0.33	0.41	0.53	0.35	0.42	0.52	0.44	0.47	0.54	0.40	0.46
5	0.57	0.31	0.40	0.58	0.33	0.42	0.58	0.42	0.48	0.58	0.37	0.45
6	0.61	0.29	0.39	0.61	0.31	0.41	0.60	0.39	0.47	0.61	0.34	0.44
7	0.63	0.27	0.38	0.64	0.29	0.40	0.64	0.37	0.47	0.63	0.32	0.42
8	0.66	0.26	0.37	0.67	0.28	0.39	0.67	0.36	0.47	0.65	0.31	0.42
9	0.68	0.25	0.36	0.68	0.27	0.38	0.69	0.35	0.46	0.67	0.30	0.41
10	0.70	0.24	0.35	0.70	0.26	0.38	0.71	0.34	0.46	0.69	0.29	0.41
11	0.71	0.23	0.34	0.71	0.25	0.37	0.72	0.33	0.45	0.70	0.28	0.40
12	0.73	0.22	0.34	0.73	0.24	0.37	0.74	0.32	0.45	0.71	0.28	0.40
13	0.74	0.22	0.34	0.74	0.24	0.36	0.75	0.32	0.45	0.72	0.27	0.40
14	0.76	0.21	0.33	0.76	0.24	0.36	0.77	0.32	0.45	0.74	0.27	0.40
15	0.77	0.21	0.33	0.77	0.24	0.36	0.78	0.31	0.45	0.75	0.27	0.40
16	0.78	0.21	0.33	0.78	0.23	0.36	0.79	0.31	0.44	0.75	0.27	0.39
17	0.79	0.20	0.32	0.79	0.23	0.36	0.80	0.31	0.44	0.76	0.26	0.39
18	0.79	0.20	0.32	0.80	0.23	0.35	0.80	0.30	0.44	0.77	0.26	0.39
19	0.80	0.20	0.32	0.80	0.23	0.35	0.81	0.30	0.44	0.77	0.26	0.39
20	0.81	0.20	0.32	0.81	0.22	0.35	0.82	0.30	0.44	0.77	0.26	0.39
50	0.84	0.16	0.27	0.84	0.20	0.32	0.85	0.26	0.40	0.80	0.22	0.35

R, recall; P, precision; F, f-measure

Bibliography

1 Zika E, Paci D, Braun a., *et al.* A European survey on biobanks: Trends and issues. *Public Health Genomics* 2011;**14**:96–103. doi:10.1159/000296278

2 De Souza YG, Greenspan JS. Biobanking past, present and future: responsibilities and benefits. *AIDS (London, England)* 2013;**27**:303–12. doi:10.1097/QAD.0b013e32835c1244

3 Brandsma M, Van Ommen G-JB, Wijmenga C, *et al.* Dutch government invests in existing biobanks. *Nederlands Tijdschrift Voor Geneeskunde* 2010;**154**:A2825.http://www.ncbi.nlm.nih.gov/pubmed/21029488

4 Holub P, Swertz M, Reihs R, *et al.* BBMRI-ERIC Directory: 515 Biobanks with Over 60 Million Biological Samples. *Biopreservation and Biobanking* 2016;**14**:559–62. doi:10.1089/bio.2016.0088

5 Scholtens S, Smidt N, Swertz MA, *et al.* Cohort Profile: LifeLines, a three-generation cohort study and biobank. *International journal of epidemiology* 2015;**44**:1172–80. doi:10.1093/ije/dyu229

6 Van Vliet-Ostaptchouk J V, Nuotio M-L, Slagter SN, *et al.* The prevalence of Metabolic Syndrome and metabolically healthy obesity in Europe: a collaborative analysis of ten large cohort studies. *BMC Endocrine Disorders* 2014;**14**:1–13. doi:10.1186/1472-6823-14-9

7 Pathak J, Wang J, Kashyap S, *et al.* Mapping clinical phenotype data elements to standardized metadata repositories and controlled terminologies: the eMERGE Network experience. *Journal of the American Medical Informatics Association : JAMIA* 2011;**18**:376–86. doi:10.1136/amiajnl-2010-000061

8 Fortier I, Doiron D, Little J, *et al.* Is rigorous retrospective harmonization possible? Application of the DataSHaPER approach across 53 large studies. *International Journal of Epidemiology* 2011;**40**:1314–28. doi:10.1093/ije/dyr106

9 Weidman S, Arrison T. *Steps Toward Large-Scale Data Integration in*

the Sciences. Washington, D.C.: : National Academies Press 2010. doi:10.17226/12916

10 Norlin L, Fransson MN, Eriksson M, *et al.* A Minimum Data Set for Sharing Biobank Samples, Information, and Data: MIABIS. *Biopreservation and Biobanking* 2012;**10**:343–8. doi:10.1089/bio.2012.0003

11 Maelstrom Research. Maelstrom Research. 2015.https://www.maelstrom-research.org/ (accessed 9 Mar2017).

12 Euzenat J, Shvaiko P. *Ontology Matching*. Second. 2013. http://www.springer.com/computer/database+management+&+information+retrieval/book/978-3-642-38720-3

13 Vassiliadis P. A survey of Extract – transform – Load technology. *International Journal of Data Warehousing & Mining* 2009;**5**:1–27.http://bit.ly/15KE6p1

14 Ziegler P, Dittrich KR. Three Decades of Data Integration — All Problems Solved? *In 18th IFIP World Computer Congress (WCC 2004),* 2004;**12**:3–12. doi:10.1007/b98986

15 Gaye A, Marcon Y, Isaeva J, *et al.* DataSHIELD: Taking the analysis to the data, not the data to the analysis. *International Journal of Epidemiology* 2014;**43**:1929–44. doi:10.1093/ije/dyu188

16 Paz-Trillo C, Wassermann R, Braga PP. An information retrieval application using ontologies. *Journal of the Brazilian Computer Society* 2005;**11**:17–31. doi:10.1007/BF03192373

17 Rodd J, Gaskell G, Marslen-Wilson W. Making Sense of Semantic Ambiguity: Semantic Competition in Lexical Access. *Journal of Memory and Language* 2002;**46**:245–66. doi:10.1006/jmla.2001.2810

18 Magma. Magma Javascript API. 2014.http://wiki.obiba.org/display/OPALDOC/Magma+Javascript+API (accessed 20 Jul2015).

19 Opal. Opal. 2011.http://www.obiba.org/pages/products/opal/ (accessed

20 Jul2015).

20 Doiron D, Burton P, Marcon Y, *et al.* Data harmonization and federated analysis of population-based studies: the BioSHaRE project. *Emerging Themes in Epidemiology* 2013;**10**:12. doi:10.1186/1742-7622-10-12

21 Pathak J, Wang J, Sudha MS, *et al.* eleMAP : An Online Tool for Harmonizing Data Elements using Standardized Metadata Registries and Biomedical Vocabularies. *AMIA . Annual Symposium proceedings / AMIA Symposium AMIA Symposium* Published Online First: 2010. doi:doi=10.1.1.172.7082

22 Burdett T, Jupp S, Malone J, *et al.* Zooma2 - A repository of annotation knowledge and curation API. 2012.http://www.ebi.ac.uk/spot/zooma/index.html (accessed 26 Jun2015).

23 Gostev M, Fernandez-Banet J, Rung J, *et al.* SAIL--a software system for sample and phenotype availability across biobanks and cohorts. *Bioinformatics (Oxford, England)* 2011;**27**:589–91. doi:10.1093/bioinformatics/btq693

24 Athey BD, Braxenthaler M, Haas M, *et al.* tranSMART: An Open Source and Community-Driven Informatics and Data Sharing Platform for Clinical and Translational Research. *AMIA Joint Summits on Translational Science proceedings AMIA Summit on Translational Science* 2013;**2013**:6–8.http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3814495&tool=pmcentrez&rendertype=abstract

25 Fortier I, Burton PR, Robson PJ, *et al.* Quality, quantity and harmony: The DataSHaPER approach to integrating data across bioclinical studies. *International Journal of Epidemiology* 2010;**39**:1383–93. doi:10.1093/ije/dyq139

26 Abbasi A, Corpeleijn E. External validation of the KORA S4/F4 prediction models for the risk of developing type 2 diabetes in older adults: the PREVEND study. *European Journal of Epidemiology*

2012;**27**:47–52. doi:10.1007/s10654-011-9648-4

27 Aleksovski Z, Klein M, Ten Kate W, *et al.* Matching Unstructured Vocabularies using a Background Ontology. *Lecture Notes in Computer Science* 2006;**4248**:182–97. doi:10.1007/11891451_18

28 GIUNCHIGLIA F, SHVAIKO P. Semantic matching. The Knowledge Engineering Review. 2003;**18**:265–80. doi:10.1017/S0269888904000074

29 Rubin DL, Shah NH, Noy NF. Biomedical ontologies: a functional perspective. *Briefings in Bioinformatics* 2008;**9**:75–90.

30 Díaz-Galiano MC, Martín-Valdivia MT, Ureña-López LA. Query expansion with a medical ontology to improve a multimodal information retrieval system. *Computers in Biology and Medicine* 2009;**39**:396–403. doi:10.1016/j.compbiomed.2009.01.012

31 Doms A, Schroeder M. GoPubMed: exploring PubMed with the Gene Ontology. *Nucleic Acids Research* 2005;**33**:W783–6. doi:10.1093/nar/gki470

32 Ashburner M, Ball CA, Blake JA, *et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nature genetics* 2000;**25**:25–9. doi:10.1038/75556.Gene

33 Rodriguez MDB, Hidalgo JMG, Agudo BD. Using WordNet to Complement Training Information in Text Categorization. In: *Recent Advances in Natural Language Processing II Selected Papers from the Second International Conference on Recent Advances in Natural Language Processing RANLP 1997 March 2527 1997 Stanford CA USA*. 1997. 16.

34 Kristina Nilsson, Hans Hjelm HO. SUiS – cross-language ontology-driven information retrieval in a restricted domain. In: *In Proceedings of the 15th NODALIDA conference*. 2005.

35 Voorhees EM. Using WordNet to disambiguate word senses for text retrieval. In: *Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*

SIGIR 93. 1993. 171–80. doi:10.1145/160688.160715

36 Ehrig M. Foam - framework for ontology alignment and mapping; results of the ontology alignment initiative. In: *Proceedings of the Workshop on Integrating Ontologies. Volume 156., CEUR-WS.org (2005) 72–76*. 2005. 72–6.

37 Giunchiglia F, Autayeu A, Pane J. S-match: an open source framework for matching lightweight ontologies. *Semant web* 2012;**3**:307–17. doi:10.3233/SW-2011-0036

38 Clinical Information Modeling Initiative (CIMI). http://informatics.mayo.edu/CIMI/index.php/Main_Page

39 Data Standards Registry and Repository (caDSR). http://cbiit.nci.nih.gov/ncip/biomedical-informatics-resources/interoperability-and-semantics/metadata-and-models

40 Swertz MA, Dijkstra M, Adamusiak T, *et al.* The MOLGENIS toolkit: rapid prototyping of biosoftware at the push of a button. *BMC Bioinformatics* 2010;**11**:S12. doi:10.1186/1471-2105-11-S12-S12

41 Adamusiak T, Parkinson H, Muilu J, *et al.* Observ-OM and Observ-TAB: Universal syntax solutions for the integration, search and exchange of phenotype and genotype information. *Human Mutation* 2012;**33**:867–73. doi:10.1002/humu.22070

42 Whetzel PL, Shah NH, Noy NF, *et al.* BioPortal : Ontologies and Integrated Data Resources at the Click of a Mouse. *Nucleic acids research* 2009;**37**:170–3.

43 P3G_Observatory. P3G Observatory. 2005.www.p3gobservatory.org

44 Adamusiak T, Burdett T, Kurbatova N, *et al.* OntoCAT -- simple ontology search and integration in Java, R and REST/JavaScript. *BMC Bioinformatics* 2011;**12**:218. doi:10.1186/1471-2105-12-218

45 The Apache Software Foundation. Apache Lucene. Agenda. 2006;**2009**.https://lucene.apache.org/core/

46 Wolffenbuttel B. Healthy Obese Project.

2013;:1.https://www.bioshare.eu/content/healthy-obese-project

47 Diercks GF, Van Boven AJ, Hillege HL, *et al.* Microalbuminuria is independently associated with ischaemic electrocardiographic abnormalities in a large non-diabetic population. The PREVEND (Prevention of REnal and Vascular ENdstage Disease) study. *European Heart Journal* 2000;**21**:1922–7.

48 Mao M, Peng Y, Spring M. An adapatative ontology mapping approach with neural network based constraint satisfaction. *Web Semantics: Services and Agents on the World Wide Web* 2010;**8**:14–25.

49 BioShaRE. BioSHaRE project. 2011.https://www.bioshare.eu/ (accessed 6 Jun2015).

50 Pang C, Hendriksen D, Dijkstra M, *et al.* BiobankConnect: software to rapidly connect data elements for pooled analysis across biobanks using ontological and lexical indexing. *Journal of the American Medical Informatics Association* 2015;**22**:65–75.http://jamia.oxfordjournals.org/content/22/1/65.abstract

51 Poggi A, Lembo D, Calvanese D, *et al.* Linking data to ontologies. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer 2008. 133–73. doi:10.1007/978-3-540-77688-8_5

52 Rubin DL, Shah NH, Noy NF. Biomedical ontologies: A functional perspective. *Briefings in Bioinformatics* 2008;**9**:75–90. doi:10.1093/bib/bbm059

53 Zwamborn-Hanssen AMN, Bijlsma JB, Hennekam EFAM, *et al.* The Dutch uniform multicenter registration system for genetic disorders and malformation syndromes. *American Journal of Medical Genetics* 1997;**70**:444–7. doi:10.1002/(SICI)1096-8628(19970627)70:4<444::AID-AJMG20>3.0.CO;2-G

54 Navarro G. A guided tour to approximate string matching. ACM Computing Surveys. 2001;**33**:31–88. doi:10.1145/375360.375365

55 Brown PF, DeSouza P V., Mercer RL, *et al.* Class-Based n-gram

Models of Natural Language. *Computational Linguistics* 1992;**18**:467–79.http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.13.9919%5Cnhttp://acl.ldc.upenn.edu/J/J92/J92-4003.pdf

56 Levenshtein VI. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady* 1966;**10**:707–10.

57 Salton G, Wong A, Yang CS. A vector space model for automatic indexing. *Communications of the ACM* 1975;**18**:613–20. doi:10.1145/361219.361220

58 Saruladha K. A Comparative Analysis of Ontology and Schema Matching Systems. *International Journal of Computer Applications* 2011;**34**:14–21.

59 Mathur I, Joshi N. Shiva ++ : An Enhanced Graph based Ontology Matcher. *International Journal of Computer Applications* 2014;**92**:30–4.

60 Cruz IF. AgreementMaker : Efficient Matching for Large Real-World Schemas and Ontologies. *PVLDB* 2009;**2**:1586–9. doi:10.14778/1687553.1687598

61 Jiménez-Ruiz E, Cuenca Grau B. LogMap: Logic-based and scalable ontology matching. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer 2011. 273–88. doi:10.1007/978-3-642-25073-6_18

62 Schuemie MJ, Jelier R, Kors J. Peregrine: Lightweight gene name normalization by dictionary lookup. In: *Proc of the Second BioCreative Challenge Evaluation Workshop*. 2007. 131–3.

63 Funk C, Baumgartner W, Garcia B, *et al.* Large-scale biomedical concept recognition: an evaluation of current automatic annotators and their parameters. *BMC bioinformatics* 2014;**15**:59. doi:10.1186/1471-2105-15-59

64 Apache Software Foundation. Lucene Similarity Score. 2001.https://lucene.apache.org/core/4_6_0/core/overview-summary.html (accessed 3 Jun2015).

65 ElasticSearch. ElasticSearch: Lucene’s Practical Scoring Function. 2015.https://www.elastic.co/guide/en/elasticsearch/guide/master/practical-scoring-function.html#query-norm (accessed 3 Jun2015).

66 Ainsworth BE, Haskell WL, Leon AS, *et al.* Compendium of physical activities: Classification of energy costs of human physical activities. In: *Medicine and Science in Sports and Exercise*. 1993. 71–80. doi:10.1249/00005768-199301000-00011

67 Sollie A, Sijmons RH, Lindhout D, *et al.* A New Coding System for Metabolic Disorders Demonstrates Gaps in the International Disease Classifications ICD-10 and SNOMED-CT, Which Can Be Barriers to Genotype-Phenotype Data Sharing. *Human Mutation* 2013;**34**:967–73. doi:10.1002/humu.22316

68 Sernadela, Pedro and Horst, Eelke and Thompson, Mark and Lopes, Pedro and Roos, Marco and Oliveira J. A Nanopublishing Architecture for Biomedical Data. In: *8th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACBB 2014)*. 2014. 277–84. doi:10.1007/978-3-319-07581-5_33

69 Szalma S, Koka V, Khasanova T, *et al.* Effective knowledge management in translational medicine. *Journal of Translational Medicine* 2010;**8**:68. doi:10.1186/1479-5876-8-68

70 Murphy SN, Weber G, Mendis M, *et al.* Serving the enterprise and beyond with informatics for integrating biology and the bedside (i2b2). *Journal of the American Medical Informatics Association : JAMIA* 2010;**17**:124–30. doi:10.1136/jamia.2009.000893

71 Molgenis. EMX upload format. 2014.https://github.com/molgenis/molgenis/wiki/EMX-upload-format (accessed 20 Jul2015).

72 Bhogal J, Macfarlane A, Smith P. A review of ontology based query expansion. *Information Processing & Management* 2007;**43**:866–86. doi:10.1016/j.ipm.2006.09.003

73 JScience. JScience. 2012.http://jscience.org/ (accessed 8 Jul2015).

74 Schadow G, McDonald CJ. The Unified Code for Units of Measure (UCUM). 2005.http://unitsofmeasure.org/trac

75 JSON.org. Introducing JSON. json.org. 2014.http://www.json.org/ (accessed 25 Sep2015).

76 Wu Z, Palmer M. Verb Semantics and Lexical Selection. *32nd annual meeting on Association for Computational Linguistics* 1994;:6. doi:10.3115/981732.981751

77 Shima H. WordNet similarity for Java. 2011.https://code.google.com/p/ws4j/ (accessed 4 Nov2015).

78 Pang C, Sollie A, Sijtsma A, *et al.* SORTA: a system for ontology-based re-coding and technical annotation of biomedical phenotype data. *Database* 2015;**2015**:bav089. doi:10.1093/database/bav089

79 Pang C, van Enckevort D, de Haan M, *et al.* MOLGENIS/connect: a system for semi-automatic integration of heterogeneous phenotype data with applications in biobanks. *Bioinformatics* 2016;**32**:btw155-. doi:10.1093/bioinformatics/btw155

80 Miles A, Pérez-Agüera JR. SKOS: Simple Knowledge Organisation for the Web. *Cataloging & Classification Quarterly* 2007;**43**:69–83. doi:Article

81 Pennington J, Socher R, Manning CD. GloVe: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing* 2014;:1532–43. doi:10.3115/v1/D14-1162

82 Pang C, Hendriksen D, Dijkstra M, *et al.* BiobankConnect: software to rapidly connect data elements for pooled analysis across biobanks using ontological and lexical indexing. *Journal of the American Medical Informatics Association* 2014;:65–75. doi:10.1136/amiajnl-2013-002577

83 Manning CD, Bauer J, Finkel J, *et al.* The Stanford CoreNLP Natural Language Processing Toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations* 2014;:55–60. doi:10.3115/v1/P14-5010

84 Hendler J. Data Integration for Heterogenous Datasets. *Big Data* 2014;**2**:205–15. doi:10.1089/big.2014.0068

85 Wilkinson MD, Dumontier M, Aalbersberg IjJ, *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 2016;**3**:160018. doi:10.1038/sdata.2016.18

86 Bianchi S, Burla A, Conti C, *et al.* Biomedical data integration - Capturing similarities while preserving disparities. In: *Proceedings of the 31st Annual International Conference of the IEEE Engineering in Medicine and Biology Society: Engineering the Future of Biomedicine, EMBC 2009*. 2009. 4654–7. doi:10.1109/IEMBS.2009.5332650

87 Scheufele E, Aronzon D, Coopersmith R, *et al.* tranSMART: An Open Source Knowledge Management and High Content Data Analytics Platform. *AMIA Summits on Translational Science Proceedings* 2014;**2014**:96–101.http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4333702/

88 Ashish N, Dewan P, Toga AW. The GAAIN Entity Mapper: An Active-Learning System for Medical Data Mapping. *Frontiers in neuroinformatics* 2015;**9**:30. doi:10.3389/fninf.2015.00030

89 Günther F, Fritsch S. neuralnet: Training of Neural Networks. *The R Journal* 2010;**2**:30–8.

Summary

Biobanks and patient registries provide indispensable human data for biomedical research and for the translation of these research findings into healthcare. In recent years, biomedical research has expanded dramatically from an interest in simple traits to a focus on complex multifactorial disorders where many genetic and environmental factors need to be taken into consideration to understand the underlying mechanism of development of diseases. These studies now require vast datasets to reach sufficient statistical power to make new discoveries, e.g. in the case of complex diseases where many small contributing factors add up to disease risk or the case of rare diseases or phenotypes with low prevalence where many small cohorts need to be pooled to reach sufficient numbers of affected patients. Therefore, in many cases, data from multiple biobank repositories must be pooled to enable integrated analysis, a difficult and time-intensive process that includes many technical and ethical/legal challenges. In one example of this kind of application, the EU BioSHARE consortium took four years to pool data from 15 biobanks to understand why some obese individuals remain healthy while the majority develop specific obesity-related health problems.

In this thesis we address the challenge of pooling 'phenotypic' data across multiple biobanks, i.e. how to combine the observable characteristics of many individuals that are often collected using very different questionnaires. Most biobanks contain over 1000 phenotypic data items on aspects such as demographics, lifestyle, environment and disease, and there are at least 1400 known biobanks in Europe that each have their own, unique data item collections. Our main focus has been on resolving the difficulties faced in discovering relevant datasets and their data items (data discovery), mapping relevant data items in large and complex heterogeneous datasets to one comparable standard so they can potentially be analysed in unison (data harmonization), and finally converting these heterogeneous data into one dataset that is ready for integrated analysis (data integration). The result is a set of novel computational methods and usable software implementations that efficiently resolve the differences in data capture and description among data

repositories so that researchers can quickly discover relevant data, then harmonize and integrate this data for pooled analyses. The core concept in each method is the use of ontologies, which are structured representation of biomedical knowledge, to disambiguate the semantics of metadata and data values in each dataset. These ontologies are combined with smart methods for comparison of textual descriptions (lexical matching) in order to semi-automatically pool equivalent information about standard data elements from multiple datasets into a standard common data model for downstream analyses.

We broke down the data integration pipeline into three individual tasks: matching source data elements to the standard data elements (BiobankConnect, Chapter 2), standardizing data values using ontology-matching algorithms (SORTA, Chapter 3), and assisting users in semi-automatically generating data transformation algorithms to integrate data from source values into a common data model (MOLGENIS/connect, Chapter 4). Finally, in Chapter 5, we use the same methodology to facilitate research-question-driven data discovery by piloting a method and software application that we named 'BiobankUniverse'. Below we summarize each chapter individually.

Chapter 2 describes the BiobankConnect method we developed to disambiguate data elements from different datasets using synonym and hierarchical relationship information extracted from ontologies. In BiobankConnect we incorporated advanced indexing technology (lexical matching) to generate potential matches between source and standard data elements based on relevance scores. Using this method, researchers can quickly determine the harmonization potential of each source biobank for matching a target 'common' data model that contains the research variables required to answer a specific research question (instead of manually browsing through the typically thousands of data items available).

Chapter 3 describes the SORTA method we developed to standardize data values captured in free text format as a basis for integrated analysis. Here we implemented an enhanced version of the n-gram matching algorithm incorporating TF-IDF (Term Frequency Inverse-Document Frequency) to match free text to ontology terms. More importantly, we designed SORTA to allow local terminologies (which have been stored in formats other than the standard ontology formats) to be uploaded to the system to facilitate the local standardization.

Chapter 4 describes the MOLGENIS/connect application, a semi-automatic pipeline to extract data from sources, transform data according to standard definitions, and finally load data into the common data model. In MOLGENIS/connect we have added smart functions to assist users in creating data transformation scripts such as semantic search for matching source data elements, automatic unit conversion of source data values (e.g. meter to centimeter), and automatic matching of text-based categorical data elements between sources and the standard data common model (e.g. ‘male’ to ‘M’). This method has been used beyond this PhD thesis in biobank consortia such as BBMRI-ERIC and RD-Connect.

Chapter 5 describes BiobankUniverse, where we implemented a new data discovery method for biobank data. Researchers and biobankers can upload data dictionaries containing lists of data elements in their biobank repositories as well as data common models to the same metadata network. We then automatically annotate all data dictionaries using UMLS ontology terms and, based on these, compute semantic similarities scores that represent the distances of the data repositories in BiobankUniverse. Using this network we can quickly discover similar datasets that cluster together.

In each of the chapters we performed rigorous evaluation of the new methods we developed, and in all cases found unexpectedly high precision, high recall and—most importantly—the potential for a dramatic reduction in the data integration work. Interestingly, during this thesis writing, the topic of data integration became a focus of attention, in particular with the uptake by researchers in the biobanking community of the FAIR principles of Findability, Accessibility, Interoperability and Reusability. We are convinced that the computational methods developed in this thesis can greatly help to retrospectively 'FAIR-ify' research data, i.e. make existing data adhere to

FAIR principles. In addition we have witnessed recent mainstreaming of machine learning methods. While not yet published beyond this thesis, our first experiments using these methods as basis for data item classification look very promising.

In conclusion, in this thesis we have demonstrated new computational methods to reduce barriers to data discovery, harmonization and integration. We have further demonstrated that implementation of these methods in user friendly tools can free researchers from most of the manual effort and time burden of data transformation or data discovery and can allow them to focus on answering research questions. We hope our work will further enable 'FAIR' data reuse to improve scientific efficiency and reproducibility, and that these will speed advances that ultimately inform patient care and healthy aging.

Samenvatting

Grote gegevensverzamelingen rondom menselijke proefpersonen/patiënten, zoals biobanken en patiënten registraties, zijn onmisbaar geworden voor onderzoek naar ziekte en gezondheid, en de vertaling van dit onderzoek naar zorg en preventie. De afgelopen jaren heeft dit soort onderzoek een enorme vlucht genomen, van beperkte studies in context van specifieke ziektebeelden tot nu grootschalig bestuderen van ziekten en het complexe samenspel van genetische en omgevingsfactoren. Succesvolle uitvoering van dit soort studies vereist enorme datasets, in het geval van complexe ziekten om voldoende statistische 'power' te verkrijgen en in het geval van zeldzame ziekten om voldoende patiënten te vinden. Aangezien de meeste bestaande verzamelingen per stuk (te) klein zijn, en het ook niet realistisch is om nieuwe studies te starten met miljoenen deelnemers, zal meer en meer data van meerdere biobanken moeten worden gecombineerd als basis voor een geïntegreerde analyse. Doordat de data in biobanken typisch is verzameld voor verschillende doelen, en daardoor dus ook qua structuur en samenstelling verschillen, is data integratie een moeizaam en tijdsintensief proces waarbij vele methodologische, technische en ethisch/juridische horden moeten worden genomen. Een goed voorbeeld is het EU BioSHaRE consortium waarbij gedurende een project van 4 jaar data van meer dan 15 biobanken is gecombineerd om te begrijpen waarom sommige mensen met obesitas gezond blijven terwijl de meesten allerlei ziekten ontwikkelen.

Dit proefschrift beschrijft het onderzoek naar de uitdagingen rondom het 'poolen' van phenotypische gegevens over duizenden personen in meerdere biobanken, waarmee we bijvoorbeeld demografie, levensstijl, omgeving en ziekte data bedoelen die typisch wordt verzameld door middel van verschillende vragenlijsten. De meeste biobanken verzamelen elk meer dan 1000 van zulke kenmerken voor elk proefpersoon en er zijn zeker meer dan 1400 van zulke biobanken in Europa die elk onderling in hoge mate verschillen. In het bijzonder hebben we ons bezig gehouden met de vraagstukken rondom (i) het effectief in kaart brengen en vindbaar maken van relevante datasets en de bijbehorende data items (data discovery), (ii) het kunnen vaststellen welke van de data items vanuit elke bron dataset potentieel gecombineerd kunnen worden als basis voor analyse (data harmonisatie) en (iii) op welke wijze deze data efficiënt kunnen worden getransformeerd naar een gestandaardiseerde dataset om daadwerkelijk geïntegreerde analyse mogelijk te maken (data integratie). Het resultaat is een collectie nieuwe computationele methoden, inclusief bruikbare software, waarmee (semi)automatisch en efficiënt verschillen in data verzameling en beschrijving kunnen worden overbrugd zodat onderzoekers veel sneller dan hiervoor data kunnen vinden, harmoniseren en integreren. De kern van deze methoden is het gebruik van gestructureerde kennis representaties, ‘ontologieën’ genaamd, waarbij voor veel van de gebruikte termen is vastgelegd hoe ze zich tot elkaar verhouden. Denk hierbij aan synoniemen, bijzondere gevallen, generalisaties, etc (bijvoorbeeld: bier, wijn, en jenever drinken is een bijzonder geval van alcohol gebruik). Deze ontologieën zijn gecombineerd met technieken voor het vergelijken van beschrijvingen (lexical matching) om zo de enorme zoekopdracht van het vinden en op elkaar projecteren van wetenschappelijke data items te kunnen automatiseren.

In dit proefschrift hebben we de data integratie pipeline opgedeeld in drie taken: het vinden van welke data items in elke databron passen op een set ‘standaard’ data items die nodig is om de onderzoeksvraag te beantwoorden (BiobankConnect, Hoofdstuk 2), het opschonen van de bron data daar waar men vrije tekst beschrijvingen of non-standaard categorieën gebruikt (SORTA, Hoofdstuk 3), en een semi-automatische procedure om daadwerkelijk data uit de verschillende bronnen te transformeren in een standaard data model klaar voor geïntegreerde analyse (MOLGENIS/connect, Hoofdstuk 4). Tenslotte beschrijven we in Hoofdstuk 5 hoe we deze technologieën ook hebben gebruikt om een zoekmachine te maken, genaamd ‘BiobankUniverse’, waarmee onderzoekers snel kunnen vinden in hoeverre biobanken de benodigde gegevens bevatten. Hieronder een korte beschrijving van elk hoofdstuk.

Hoofdstuk 2 beschrijft de nieuwe BiobankConnect methode waarin met behulp van kennis omtrent synoniemen en hiërarchische relaties de vaak heel verschillende beschrijvingen van data items met elkaar in lijn kunnen worden gebracht zodat kan worden vastgesteld of ze gezamenlijk geanalyseerd kunnen worden. Deze methode maakt gebruikt van geavanceerde indexeer technologie (lexical matching) om voor elke gewenste onderzoeksvariabele een lijst van kandidaat ‘matches’ te genereren. Zodoende hoeven onderzoekers niet met de hand alle duizenden data items bij langs maar kan snel worden beoordeeld in hoeverre elke databron de benodigde data items bevat.

Hoofdstuk 3 beschrijft de SORTA methode waarmee vrije tekst (uit bijvoorbeeld open vragen in vragenlijsten) efficiënt kan worden ‘gecodeerd’ in standaardbepalingen wat nodig is voordat statistische analyse kan plaatsvinden. In deze methode hebben we een verbeterde versie van het ‘n-gram’ algoritme ontwikkeld om vrije tekst te kunnen koppelen aan ontologie termen (met behulp van TF-IDF, Term Frequency Inverse-Document Frequency). Daarnaast kan SORTA ook gekoppeld worden aan niet-ontologische codesystemen/categorie systemen zodat ook geconverteerd kan worden naar lokale standaarden.

Hoofdstuk 4 beschrijft de MOLGENIS/connect pipeline waarmee data vanuit de bronbestanden semi-automatisch kan worden getransformeerd naar de gewenste standaard. Het systeem ‘raadt’ automatisch welk data transformatie algoritmes waarschijnlijk noodzakelijk zijn om de brondata om te zetten. Hiervoor is de BiobankConnect methode voor ‘matching’ uitgebreid om automatisch data transformatie scripts voor eenheden conversies te genereren (bijvoorbeeld van meter naar centimeter) en de SORTA methode voor categorie conversie uit te breiden voor het genereren van scripts voor categorie conversie (bijvoorbeeld ‘male’ to ‘M’). Een menselijke expert kan vervolgens deze scripts controleren en vervolgens toepassen om daadwerkelijk de data vanuit meerdere bronnen in een dataset samen te brengen. Deze pipeline wordt nu in productie gebruikt voorbij de toepassingen beschreven in dit proefschrift in biobank consortia BBMRI-ERIC en RD-Connect.

Hoofdstuk 5 beschrijft BiobankUniverse waarin we een nieuwe methode hebben ontwikkeld voor het kunnen vinden van data in biobanken. Als biobankiers/onderzoekers de complete definitie van al hun data items uploaden in BiobankUniverse dan worden deze automatisch geclassificeerd tegen de UMLS ontologie. Vervolgens wordt op basis van deze classificatie een semantische gelijkenis score uitgerekend waarmee een maat voor de ‘afstand’ tussen gehele data collecties alsook individuele data items is gerealiseerd. Op basis van deze maat kan zeer snel gegeven een zoekvraag, bijvoorbeeld ‘hartziekten’, gelijksoortige gegevens worden opgevraagd.

Elk van de methoden is grondig geëvalueerd in de context van praktijkvoorbeelden en in alle gevallen vonden we een hoge precisie en opbrengst en - vooral van belang - een grote vermindering van het menselijk handwerk benodigd voor data integratie. Daarnaast stellen wij met blijdschap vast dat de interesse in de vraagstukken rondom data integratie en hergebruik de afgelopen jaren enorm is toegenomen. Dit is mede te danken aan wereldwijd draagvlak voor de gedachte dat alle wetenschappelijke data ‘FAIR’ zou moeten zijn, waarmee bedoeld wordt: vindbaar, toegankelijk, integreerbaar en herbruikbaar (Findable, Accessible, Interoperable, Reusable). Wij zijn ervan overtuigd dat we met de computationele methoden in dit proefschrift een grote bijdrage kunnen leveren aan het ‘retrospectief’ FAIR maken van bestaande data. Daarnaast denken we dat het recent gemeengoed worden van machine learning technieken nieuwe kansen biedt om de prestaties van deze methoden nog verder te verbeteren.

Tot besluit: dit proefschrift heeft laten zien hoe nieuwe computationele methoden de barrières voor het kunnen vinden, harmoniseren en integreren/hergebruiken van bestaande data enorm kan verminderen. Daarnaast is vastgesteld dat implementatie van deze methoden in gebruiksvriendelijk software kan helpen om onderzoekers te bevrijden van langdurig handmatig ‘corvee’ werk waardoor meer tijd voor het beantwoorden van onderzoeksvragen overblijft. Wij hopen dan ook dat ons werk het mogelijk zal maken om op grote schaal data ‘FAIR’ te maken zodat de grote investeringen in wetenschappelijke data meervoudig hergebruikt kunnen worden en we daarmee een bijdrage leveren aan verbetering van patiëntenzorg en het stimuleren van gezond oud worden.

Acknowledgements

This is quite an epic journey finally coming to an end. I've spent a few good years of my life here in Groningen. Academically I've generated some decent amount of research but can't really say I am a model student. What matters to me more personally, however, is the people I have met. During my PhD years, I was lucky enough to have had great supervisors and colleagues from whom I have learned so much and taken advice. Apart from work, I've made many friends and done crazy parties/things with you all. Those memories are still very much vivid and it's as if I'd never left. Groningen is like my home and I would like to thank many people for your help and guidance.

I would like to give the special thanks to my promotor Morris. Most importantly, you are the one that introduced and brought me into this PhD program. During those years, I was given many opportunities to learn various technologies/tools and explore new research ideas. The skill set I have mastered in those years has helped me greatly in my career. Not only did you give me a lot of freedom to experiment interesting new ideas, but also you provided guidance and wisdom at the crucial moments of my research projects.

I would also like to give the special thanks to my promotor Hans for overseeing the progress of my PhD work and providing the domain expertise in the development of all methodologies. Most importantly you always helped correct the course of my research in case I got astray from the right track. I am especially grateful to you for always reminding me to focus on finishing my thesis and graduation.

I would like to thank my paranymphs/beer buddies Dennis and Freerk for organizing the defense event and the graduation party on my behalf, especially considering the technical difficulty of me not physically being in Groningen, you guys arranged everything for me and I do appreciate that!

I want to thank our editors Kate and Jackie for your ‘magic touch’ on every single chapter of my thesis, I am very grateful that you kindly pointed out my mistakes in writing. I truly learned a lot by observing track changes.

I like to thank my friends (Gromaniacs, basketball friends and others I will not list out each one of you) I’ve met in Groningen. Our friendship is definitely beyond crazy partying + drinking (although these are very essential as well!). Your company has made this journey fun and exciting!

Special thanks to my parents and my Auntie Amanda for the support in all these years. You helped shape my personality and values and made me the person I am today. I would not be where I am if it was not for your guidance and support.

Last but not least I would like to thank my wife Emma, we were in a long distance relationship for many years while I was in Groningen doing my PhD. Your support and sacrifice were beyond words. I am very much grateful that you could be at my graduation ceremony witnessing this important moment with me. Thank you very much for always being understanding, forgiving and having faith in our future.

About the author

Chao Pang was born on May 7th 1987 in Beijing, China. He obtained his bachelor's degree in Biotechnology from University of Science and Technology Beijing in 2009. He moved to the U.K. and did a master program in Bioinformatics at University of Leicester in 2010. Having realized his passion for informatics, he joined the group of Morris Swertz in 2011 at University Medical Center of Groningen in the Netherlands to start his PhD program, where he worked on novel methods and developed software in order to address the challenges in biobank data integration and harmonization. Chao left the Netherlands for the US in 2017 and joined the Department of Biomedical Informatics at Columbia University working as a Software Developer and Data Engineer.

His primary interest is to combine Natural Language Processing, Machine learning, Semantic Web and Information Retrieval technologies to develop novel methods in order to automatically discover and integrate the relevant datasets for the research questions and common data models.

List of publications

1. **Pang C**, Kelpin F, van Enckevort D ,et al. BiobankUniverse: automatic matchmaking between datasets for biobank data discovery and integration. *Bioinformatics***.** Accepted on July 22nd 2017. doi: 10.1093/bioinformatics/btx478.

2. **Pang C**, van Enckevort D, de Haan M, et al. MOLGENIS/connect: a system for semi-automatic integration of heterogeneous phenotype data with applications in biobanks. *Bioinformatics*. 2016;32(March):btw155-. doi:10.1093/bioinformatics/btw155.

3. **Pang C**, Sollie A, Sijtsma A, et al. SORTA: a system for ontology-based re-coding and technical annotation of biomedical phenotype data. *Database*. 2015;2015:bav089. doi:10.1093/database/bav089.

4. **Pang C**, Hendriksen D, Dijkstra M, et al. BiobankConnect: software to rapidly connect data elements for pooled analysis across biobanks using ontological and lexical indexing. *Journal of the American Medical Informatics Association*. 2014;65-75. doi:10.1136/amiajnl-2013-002577.

5. van Vliet-Ostaptchouk JV, Nuotio ML, Slagter SN, et al including **Pang C**. The prevalence of metabolic syndrome and metabolically healthy obesity in Europe: a collaborative analysis of ten large cohort studies. *BMC Endocrine Disorders*. 2014;14: 9–9. doi: 0.1186/1472-6823-14-9.

6. Sarntivijai S, Lin Y, Xiang Z, et al including **Pang C**. CLO: The cell line ontology. *Journal of Biomedical Semantics* . 2014;5:37. doi: 10.1186/2041-1480-5-37.

7. Adamusiak T, Parkinson H, Muilu J, et al including **Pang C**. 2012. Observ-OM and Observ-TAB: Universal Syntax Solutions for the Integration, Search, and Exchange of Phenotype And Genotype Information. *Human Mutation.* 2012;33:867–873. doi: 10.1002/humu.22070.