ADOPT BBMRI-ERIC
GRANT AGREEMENT NO. 676550

DELIVERABLE REPORT

Deliverable no	D3.5
Deliverable Title	Ontology-based toolset for mapping of the biobanking terminologies
Contractual delivery month	M18 (March 2017)
Responsible Partner	BBMRI-ERIC CS IT WP8
Author(s)	Sebastian Mate, Christina Schüttler, Katrin Bott, Christian Knell' Dennis Kadioglu, Niina Eklund, Kaisa Silander, Hans-Ulrich Prokosch

PROVISION OF A TERMINOLOGY SERVICE FOR SEMANTIC ONTOLOGY MAPPING – FIRST TOOLSET PROTOTYPE

Executive Summary

BBMRI-ERIC is a pan-European consortium with the goal to establish and operate a research infrastructure to facilitate the access to the sample collections of participating biobanks. The ADOPT BBMRI-ERIC, as part of this initiative, aims to support the implementation of BBMRI-ERIC by providing the necessary IT services.

This document is part of a series of reports regarding the deployment of a toolset for the mapping of the distinct biobanking terminologies. This deliverable (BBMRI-ERIC CS IT D8.3 / ADOPT 3.5) describes the functionalities and the development of a first toolset prototype, which is based on reusing existing tools (e.g. the Molgenis BiobankConnect tool (Pang et al. 2015a, 2016) and the Samply Metadata Repository (Kadioglu et al. 2016, Lablans et al. 2015, Storf et al. 2017) and integrating those into a data harmonisation pipeline with new developments such as a "MDR to Molgenis Converter" (MDR2MOLGENIS). The document further describes the currently proposed workflow for integrating a biobank (and its respective local data sources/data elements) through the MDR and Molgenis functionalities into the future BBMRI ERIC network.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.

Copyright notice

Document log

Issue	Date (yyyy-mm-dd)	Comment	Author/partner
D3.5_Rev1	2017-08-28	Revised: Conclusions added	Christina Schüttler
	2017-09-05	Revised:Medium/longterm recommendations added	Hans-Ulrich Prokosch
D3.5_Rev1	2017-10-05	Update on EU recognition to comply with the GA art 29.4	Outi Törnwall

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.

Table of Contents

1. Introduction

In coordination with the BBMRI-ERIC CS IT project the ADOPT project is organised in eight work packages. WP1 is responsible for the coordination of the project. This involves not only the overall management of ADOPT but also the monitoring of the project's performance. For this reason several indicators will be developed by WP2 (Data Sets), WP4 (Access) and WP7 (Rare Diseases). While WP2 will provide the first BBMRI-ERIC- wide disease cohort consisting of mapped data on colon cancer, WP7 will develop the concept for a Common Service for Rare Diseases in order to collate projects and initiatives that address this subject. WP3 and WP4 will lay the basis for these efforts. The development of an IT-Gateway (WP3) enables a close interaction between European biobanks that facilitates the trans-national cross-biobank search for suitable biospecimens and omics data (WP6). In addition, WP4 will improve the users' access to the biobank connection by granting simplified access to documents, published results and samples. Moreover, a Stakeholder Forum (WP5) will be established to encourage exchange of experience and knowledge between patients, users, investigators, and industry and to offer a platform for discussions. Eventually WP8 (Internationalisation) is in charge of expanding BBMRI-ERIC within and also beyond the European Union as well as strengthening relationships with organisations related to its activities.

Since the ADOPT BBMRI-ERIC proposal is consistently aligned with the BBMRI-ERIC Common Service IT, the CS IT will take over of the tasks of ADOPT WP3. This includes the provision of a biobank registry (Task 1), the development of IT tools regarding security architectures (Task 2) and ontology mapping (Task 3), the provision of interface Connectors (Task 4) and the application of the colon cancer use-case (Task 5).

This deliverable covers one of the specific objectives of Task 3, namely the development and deployment of a toolbox to support the mapping of data attributes/value lists, biobank meta data as well as clinical case/sample description attributes to a central core terminology.
The necessity of such a mapping toolset is due to the non-comparability of data from different sources. Every partnering institution usually refers to a particular data scheme adjusted for its own use. This proceeding might create data collections that are insufficiently described and consequently unsuitable for secondary use, e.g. future research performed by external researchers. In order to avoid this pitfall and to ensure the interoperability between data sources, a toolbox will be developed. This toolbox will provide the software components that will support the mapping of the biobanks' existing data on an agreed upon common terminology for the purpose of data harmonisation.

In the following, the report outlines the specifications and functionalities that are considered essential for the mapping/harmonisation process. Moreover, the development of the first prototype of this toolset will be described.

2. Previous work

2.1 Samply MDR

Samply is an open source software (for the largest part) that consists of multiple components (https://bitbucket.org/medinfo_mainz/). Its core components are:

- Samply.MDR: Metadata Management. The MDR follows the ISO 11179 standard (http://metadata-standards.org/11179/) and enables the definition of data elements and their associated value sets. Furthermore, data elements can be arranged in hierarchies and annotated with a rich set of attributes, such as data type or regular expression for value validation. The Samply.MDR can also be extended by making use of "slots", which allow the definition of additional arbitrary attributes. Another main feature of the MDR is its user interface. The integrated web interace allows for easy registration of new data elements, including an automatic validation of a data element's structure.
- Samply.EDC.Osse: "The OSSE registry framework is a toolbox that supports the setup of disease-related registries in the area of rare diseases. The framework consists of several local components that each registry compound runs by itself and central components that are operated for several (ideally all) OSSE-type registries." (https://bitbucket.org/medinfo_mainz/samply.edc.osse/wiki/Home)
- Samply.Auth (not open source): Central authentication within the Samply system.

For the integration into the ADOPT data harmonization processes the Samply MDR is the most relevant component (Kadioglu et al. 2016).

2.2 German Biobank Node

Within BBMRI-ERIC, it has been discussed from the very beginning to base the first prototype on the open-source software Samply (Lablans et al. 2015), a system developed for the *German Cancer Consortium* (DKTK). Samply's core functionality is to interconnect biobanks and to allow for running feasibility queries, which were created within the *Samply.Share Broker's GUI*, across the whole network. The participating biobanks are running a "bridgehead" (*Samply.Share Client*), which accepts incoming queries from the *Samply.Share Broker* (Lablans et al. 2015). For metadata management, a metadata respository, the Samply.MDR (Kadioglu et al. 2016), has been developed. Since the DKTK biobanks are based on the commercial Software CentraXX (Kairos GmbH), a CentraXX DWH Adapter has been developed. This component then issues queries to the CentraXX system.

Within the German Biobank Node (GBN) prototype, the Samply system has been successfully integrated into a hybrid querying architecture, which also comprises *Informatics for Integrating Biology and the Bedside* (i2b2) for issuing and running queries (Mate et al 2017a). As shown in Figure 1 (upper right corner), the bridgehead (*Samply.Share Client*) has been extended by a *Query Translator* and an *i2b2 DWH Adapter* (based on "li2b2" from https://github.com/li2b2) to allow for issuing queries on the i2b2 system. Since data structures and data schemas in the "Samply world" and the "i2b2 world" are different, a schema-/query translator (OmnyQery) has developed which performs an on-the-fly query translation (Mate, Vormstein et al. 2017b); in other words, the GBN network is

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.

capable of integrating biobanks with diverse data sets and different data warehouse architectures (CentraXX and i2b2). In addition, it is also possible to inject queries into the whole architecture by making use of the i2b2 webclient (upper left corner in Figure 1). This successfull pilot implementation illustrates the flexibility, but also complexity, of such hybrid networks and can be a starting point for further ADOPT/BBMRI-ERIC CS IT discussions and design decisions.

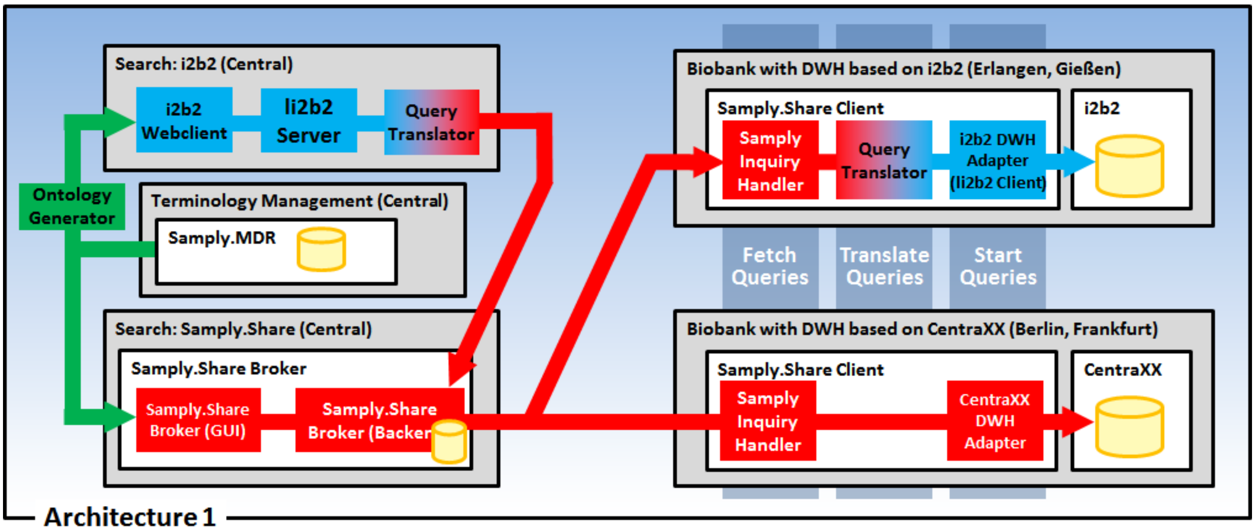


Figure 1. One of the two architectures from the German Biobank Node (GBN) prototype (modified excerpt from (Mate and Vormstein 2017a)).

The currently proposed ADOPT prototype is based on a slightly different design decision. Instead of applying a "on-the-fly" query translation, we aim at defining one core data terminology and performing the data harmonisation already beforehand as part of the ETL processes. In other words, the biobanks will have to perform the data harmonisation/integration locally, and the Connectors will store only already harmonised data. For the latter harmonisation process the analysis of already existing, and openly available tools has shown the potential of the MOLGENIS BiobankConnect lexical and semantic matching modules to provide an easy user interface to significantly speed up the biobank harmonisation process and also other forms of biomedical data Integration (Pang et al 2014).

2.3 MOLGENIS

The MOLGENIS was originally developed as a toolkit for bioinformaticians with a simple language to model biological data structures and user interfaces. At the push of a button, MOLGENIS´ generator suite automatically translates these models into feature-rich, ready-to-use web applications including database, user interfaces, exchange formats, and scriptable interfaces (Swertz et al. 2010). Since then, this toolkit has continuously expanded and especiall the MOLGENIS/connect (BiobankConnect) module may be efficiently integrated into the ADOPT harmonisation Pipeline. MOLGENIS/connect provides a semi-automatic system to find, match and pool data from different sources using ontology-based query expansion to overcome variations in terminology (Pang et al. 2016).

3. Data Harmonisation as a Major Step in the Preparation Process for the BBMRI Network Readiness

Data harmonisation is a complicated process that needs to be supported by software tools. What follows is an exemplary work flow of the harmonisation process from a biobank's point of view. It includes various steps that are ideally supported well by the selection of IT tools. However, some of the tools mentioned here are just initial proposals for which usefulness and feasibility still have to be assessed:

1. Select the variables that correspond to those variables required for the Sample Locator.
2. Collect all the necessary metadata for the required variables as it corresponds to the biobank's own data, tabulate it in the data model format required by a metadata repository (MDR).
3. Translate the data into English.
4. Upload the variable metadata into the MDR and make sure the upload went correctly using data upload report tools and by manually checking the data (→ user friendly MDR import tool, including informative report and error logs).
5. Map the local variables with the required variables stored in the MDR (→ mapping tool).
6. One variable at a time, and using the mapping tool: compare the units/options for the MDR required data set, and decide whether the variable is compatible as such or are conversion rules needed before the variable can be used. Write the conversion rule (or use suitable formula from the formula bank). Save the updated local variable metadata into your local space in the MDR. (→ tool to support transformation of variables).
7. Map the local ontologies used in the biobank's data to the supported ontologies of BBMRI-ERIC, using an ontology mapping tool (→ ontology mapping tool, includes also support for translation).
8. For local ontologies that are already mapped to the supported BBMRI-ERIC ontology, that mapping could be readily used and the links between the ontologies saved in the MDR (→ MDR ontology support tool).
9. During the individual-level data upload process, there ideally should be an automatic conversion process that converts the local ontologies to the supported ontology (→ ETL tools).
10. Set up a data warehouse software (= Connector) provided by BBMRI-ERIC CSIT onto the biobank's own server, and verify that there is proper connection between the local data warehouse, the central Sample Locator software and the central MDR.
11. Prepare the individual (or sample)–level data files for the biobank's sample collection/s.

12. Upload the individual-level data to the data warehouse/Connector using the available import tools (→ Connector import tools should at best do the transformations of the data based on the transformation rules automatically, during the import process and provide import log and error log to the user to verify the data has been properly imported, the transformations of the data properly done as specified in the MDR, and the ontology mapping correctly done).
13. Test the Sample Locator to see that the data is now available for sample availability queries.

There are already some existing data harmonisation tools, ontology mapping services, metadata repositories, and translation tools available, but there is nothing between them that would unite them. This is where 'WP8 data harmonisation toolset' would come in, working as a glue combining different softwares to operate together using the same simple data model.

4. General architechture and design

The Samply Metadata Repository (Samply MDR; https://mdr-test.ccp-it.dktk.dkfz.de; compare Kadioglu et al. 2016, Lablans et al. 2015, Storf et al. 2017) will act as the central storage for all relevant data descriptions in context of the biobank environment. The core data elements that provide the basis for all network queries are defined in the Core_Query namespace (BBMRI-ERIC Core Terminology Demonstrator (CTD)). These network queries are performed and stored by the Sample Locator. The Sample Locator is a central service, just as the MDR, and is linked to the local environment of the biobanks. Each biobank's local environment consists of a proprietary local Biobank Information Management System (BIMS) as well as a generic Connector installation. The Connector functions as the link between the external CS IT services and the local biobank. It comprises three components: (1) the MOLGENIS mapping tool (Pang et al. 2015a, Pang et al. 2016), (2) the MOLGENIS data storage and (3) the tool that fetches the queries from the Sample Locator.

(1) MOLGENIS BiobankConnect mapping tool: MOLGENIS supports the semi-automatic mapping of data elements. For this purpose it requires two input files: one file defining the original data input and second file describing the target schema on which the original data elements should be mapped. The original data input file consists of two Excel sheets: one for the data attribute metadata description and the second with the original sample/donor data. Thus, in order to obtain harmonised data, each biobank needs to create such a MOLGENIS Excel input file with its local data. It has to include both a data sheet and an attributes sheet for all entities. The attributes sheet is generated from each biobank's dedicated MDR namespace (Biobank X Namespace). Therefore the first step in a harmonisation process is, that a biobank has to create its own MDR namespace and define its own data elements and corresponding metadata in this namespace. Then this Excel input file needs to be uploaded into the MOLGENIS database. The MOLGENIS target schema input file is based on the definition of the ADOPT biobank network core data elements (still to be defined within ADOPT). All such core data elements also need to be defined in the MDR Namespace (currently BBMRI-ERIC Core Terminology Demonstrator (CTD)). Thus, the Core_Query schema is created from the MDR Core_Query Namespace and deposited in the MOLGENIS database. Subsequently, the MOLGENIS Core_Query schema can be used to map the data from the MOLGENIS biobank data sheet into the MOLGENIS database "data mapped".

(2) MOLGENIS data storage: For our purpose, we also aim to use MOLGENIS as data warehouse in order to store the mapped data by the MOLGENIS BiobankConnect mapping tool. In a broader sense, MOLGENIS can be regarded a data warehouse (DWH), as it fulfills the basic requirements

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.

of basic data integration and retrieval. It can be used to load data via the EMX format and supports mapping expressions, which allow for data transformations. The data schema can be designed in different ways, and for BBMRI-ERIC we are currently working together with WP2 to design a schema that fulfills the requirements of the BBMRI-ERIC Connector. On the retrieval side, MOLGENIS features basic functions to filter the database content from the user interface. A thorough functionality for data analysis (such as OLAP in fully-featured DHW environments) is not part of MOLGENIS; however, this is not required for BBMRI-ERIC, as the Connector will perform all data retrieval. MOLGENIS features a REST interface, which allows for querying its content. For further data analysis, the MOLGENIS database can be accessed directly, bypassing the REST interface. Future versions of the BBMRI-ERIC platform may extend MOLGENIS with another, further featured DWH or database. It has been in the discussion to use OSSE or i2b2 in future version the BBMRI-ERIC infrastructure.

(3) Tool to fetch and execute queries, as described in the ADOPT deliverable 3.6 "Biobank Connector specification and reference Connector implementation": The ultimate goal of the BBMRI-ERIC Connector is to execute queries that have been created in a central location on the Connector's DWH. By reusing the above-mentioned REST interface, incoming queries from this central location will be executed on the DWH. Similar to the GBN project, we aim to implement a translator logic that translates Samply query syntax into a syntax that conforms to MOLGENIS' query formalism. This translator will be implemented by WP2 (in close collaboration with WP8) and may build upon previous work from the GBN project (Mate et al. 2017b).

By providing those three components, it is possible to perform queries in all participating biobanks even though they describe their data elements in different ways. The Connector will only store already harmonised data. The user builds his query with the Sample Locator based on the data elements taken from the Core_Query namespace. The Connector fetches this query from the backend of the Sample Locator/Broker (component (3)) and queries each biobank's local database "data mapped" (component (1)). Since this database consists of the local biobank data mapped to the Core Terminology, the Connector receives feedback if the required data elements are available in a specific biobank and can then report it back to the Sample Locator. The General Architecture Design is illustrated in Figure 2.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.

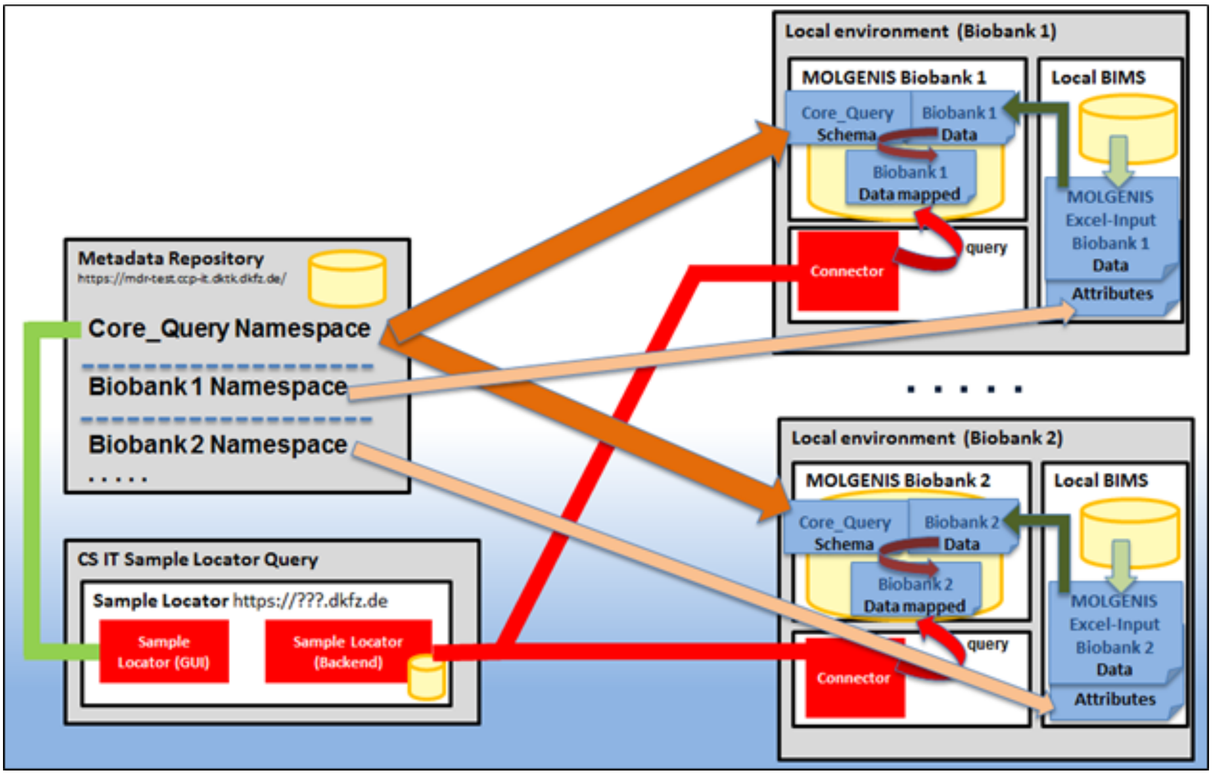


Figure 2. The General Architecture Design of the toolset.

5. Demonstrator

In order to verify the actual operability of the general architecture design, we prepared a demonstrator. Since in this first step no real biobanks and real world biobank data were involved, for testing purposes dummy datasets for a "Fake Biobank" and a "Imaginery Biobank" were created (Connector DWH mapper test.xlsx). Further, since the final definition of a core data terminology for the ADOPT / BBMRI ERIC CS IT still has to be defined within other workpackages, a demonstrator dummy core data terminology has been created (MIABIS_Sample_SampleDonor.xlsx) and defined within MDR. The procedure was as followed:

(1) Three namespaces were defined in the Demonstrator MDR (https://mdr-test.ccp-it.dktk.dkfz.de/):

(I) Namespace "BBMRI-ERIC CTD" includes several elements from MIABIS_Sample_SampleDonor.xlsx. This file consists of selected items from MIABIS 2.0, and thus will be the basis for our demonstrator by providing the core terminology.

(II) Namespace "Imaginery Biobank" includes all items from "Fake biobank data for Connector DWH mapper test.xlsx / sheet Imaginary biobank data descript". This will be the namespace of the demo biobank "Imaginery Biobank".

(III) Namespace "Fake Biobank" includes all items from "Fake biobank data for Connector DWH mapper test.xlsx / sheet Fake biobank data descript". This will be the the namespace of the second demo biobank "Fake Biobank".

(2) A parser - MDR2MOLGENIS - with a RESTful interface to the MDR was developed. The input parameter for the parser is the name of a MDR namespace. MDR2MOLGENIS reads all the data elements and their descriptions and subsequently creates a new Excel-file with an attributes sheet in accordance to the respective syntax of the MOLGENIS input files. For demonstration purpose, we run it three times with "BBMRI-ERIC CTD", "Imaginery Biobank", and "Fake Biobank" as input and create three different EMX input files. It will result in obtaining the files "CTD-Molgenis_Target Schema.emx", "Imaginery Biobank.emx", and "Fake Biobank.emx" (Figure 3).

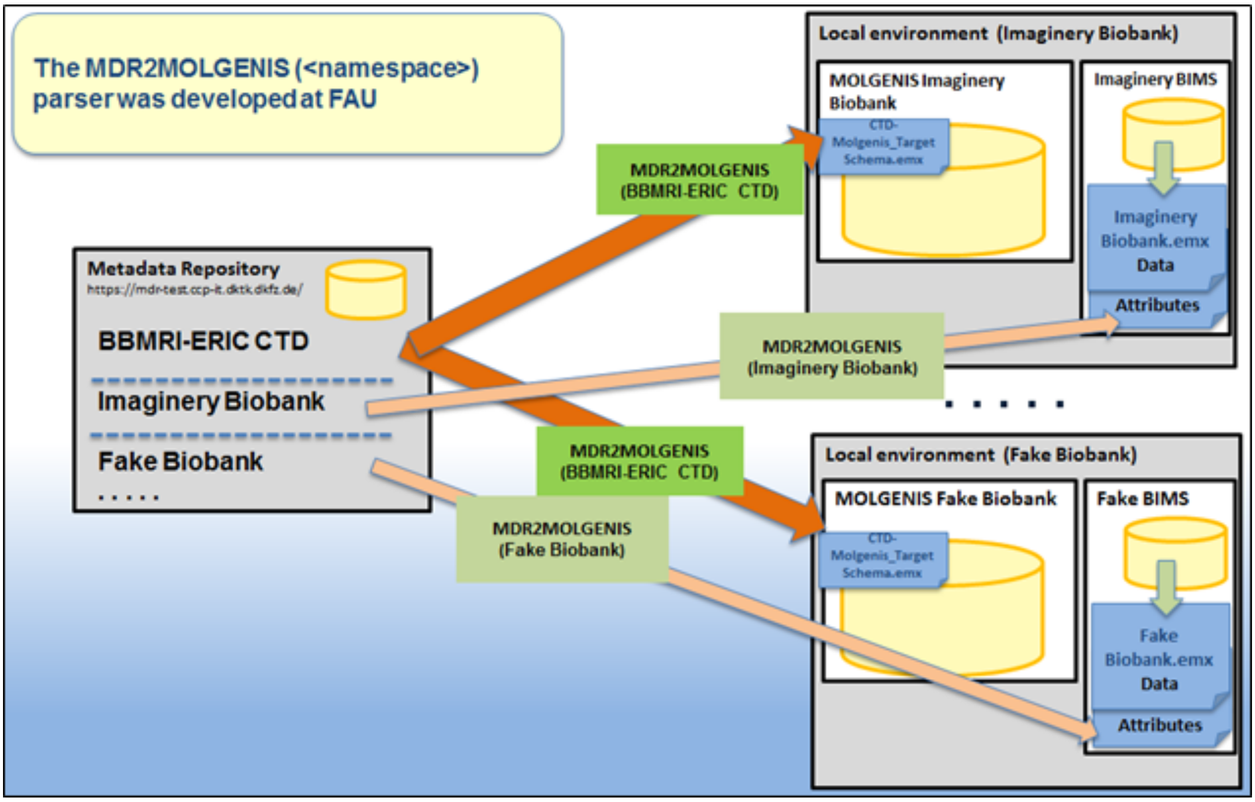

The MDR2MOLGENIS (<namespace>) parser was developed at FAU
Metadata Repository
https://mdr-test.ccp-it.dktk.dkfz.de/
BBMRI-ERIC CTD
Imaginery Biobank
Fake Biobank
. . . . .
MDR2MOLGENIS (BBMRI-ERIC CTD)
MDR2MOLGENIS (Imaginery Biobank)
MDR2MOLGENIS (BBMRI-ERIC CTD)
MDR2MOLGENIS (Fake Biobank)
Local environment (Imaginery Biobank)
MOLGENIS Imaginery Biobank
CTD-Molgenis_Target Schema.emx
Imaginery BIMS
Imaginery Biobank.emx
Data
Attributes
. . . . .
Local environment (Fake Biobank)
MOLGENIS Fake Biobank
CTD-Molgenis_Target Schema.emx
Fake BIMS
Fake Biobank.emx
Data
Attributes

Figure 3. The operating principle of MDR2MOLGENIS.

(3) The data sheet or sheets of the files "Imaginery Biobank.emx" and "Fake Biobank.emx" can subsequently be loaded with the demo "patient data" from "Fake biobank data for Connector DWH mapper test.xlsx / sheet Imaginary biobank raw data" and "Fake biobank data for Connector DWH mapper test.xlsx / sheet Fake biobank raw data" respectively.

(4) After generating those three EMX files, MOLGENIS is able to map the file "CTD-Molgenis_Target Schema.emx" with the files "Imaginery Biobank.emx" and "Fake Biobank.emx" in order to generate the database "data mapped" in the local environment of the respective biobank (Figure 4).

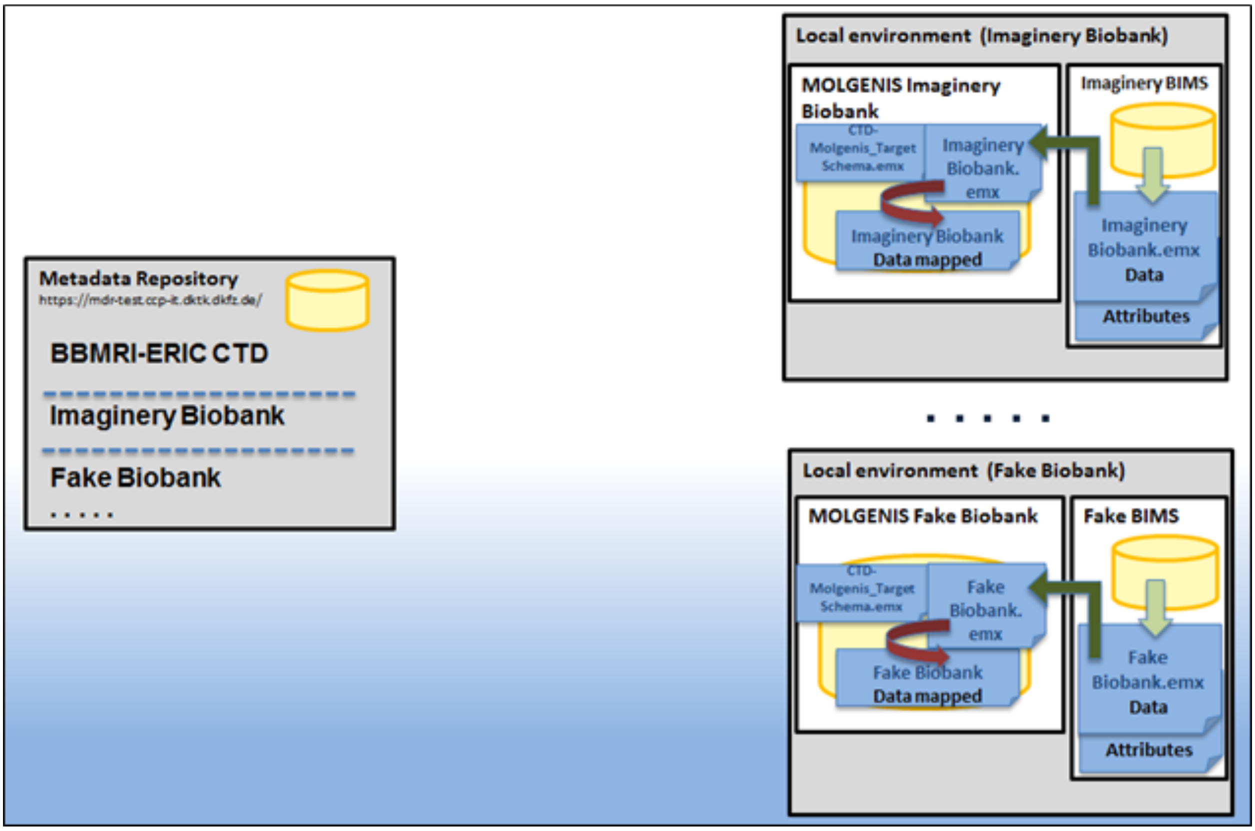

Local environment (Imaginery Biobank)
MOLGENIS Imaginery Biobank
CTD-Molgenis_Target Schema.emx
Imaginery Biobank.emx
Imaginery Biobank Data mapped
Imaginery BIMS
Imaginery Biobank.emx Data
Attributes
Metadata Repository
https://mdr-test.ccp-it.dktk.dkfz.de/
BBMRI-ERIC CTD
Imaginery Biobank
Fake Biobank
. . . . .
Local environment (Fake Biobank)
MOLGENIS Fake Biobank
CTD-Molgenis_Target Schema.emx
Fake Biobank.emx
Fake Biobank Data mapped
Fake BIMS
Fake Biobank.emx Data
Attributes

Figure 4. The mapping process of MOLGENIS in the biobank's local environment.

6. MDR2MOLGENIS

MDR2MOLGENIS is a parser, which creates the so called EMX files for using in MOLGENIS out of an MDR namespace. The RESTful interface to the MDR enables this process. Using this tool without installing anything is possible due to the integration of a Jetty server (https://eclipse.org/jetty/). Figure 5 shows the work flow of this parser.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.

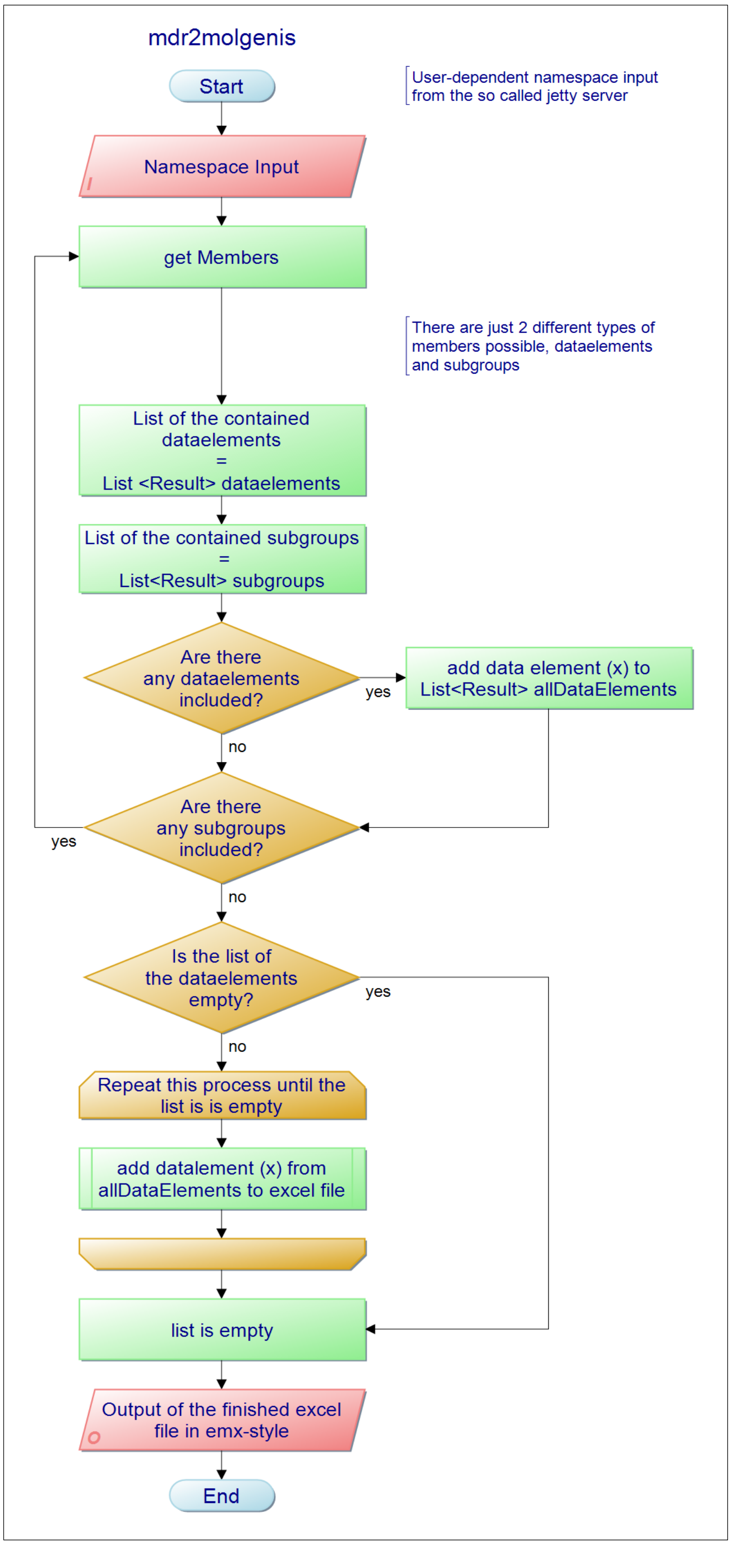

mdr2molgenis
Start
User-dependent namespace input from the so called jetty server
Namespace Input
get Members
There are just 2 different types of members possible, dataelements and subgroups
List of the contained dataelements = List <Result> dataelements
List of the contained subgroups = List<Result> subgroups
Are there any dataelements included?
yes
add data element (x) to List<Result> allDataElements
no
Are there any subgroups included?
yes
no
Is the list of the dataelements empty?
yes
no
Repeat this process until the list is is empty
add datalement (x) from allDataElements to excel file
list is empty
Output of the finished excel file in emx-style
End

Figure 5. The MDR2MOLGENIS parser algorithm.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.

The name of the desired namespace is required to be specified by the user. After receiving the input, the program retrieves the namespace members by a function as outlined below. As it can be seen, the programme differentiates the namespace into subgroups and data elements (Figure 6).

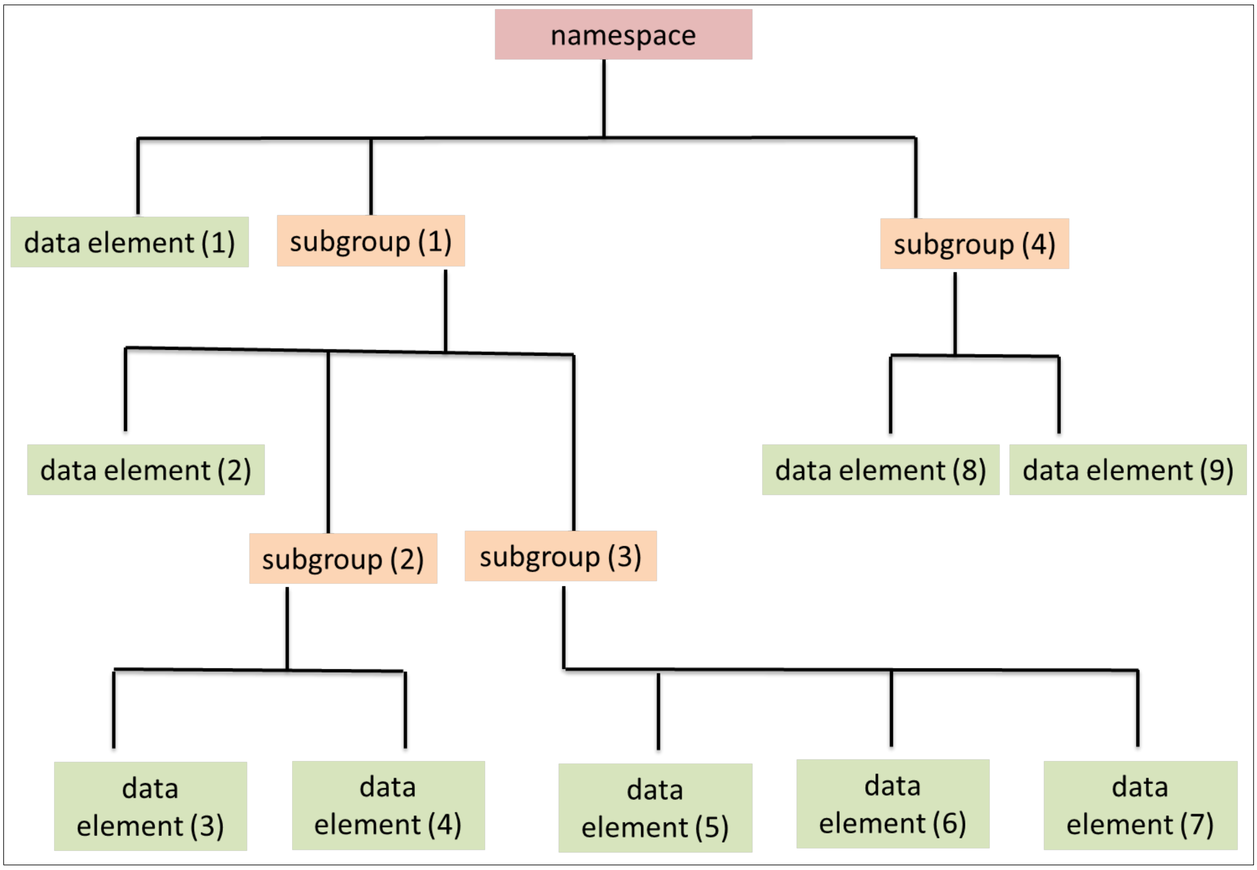
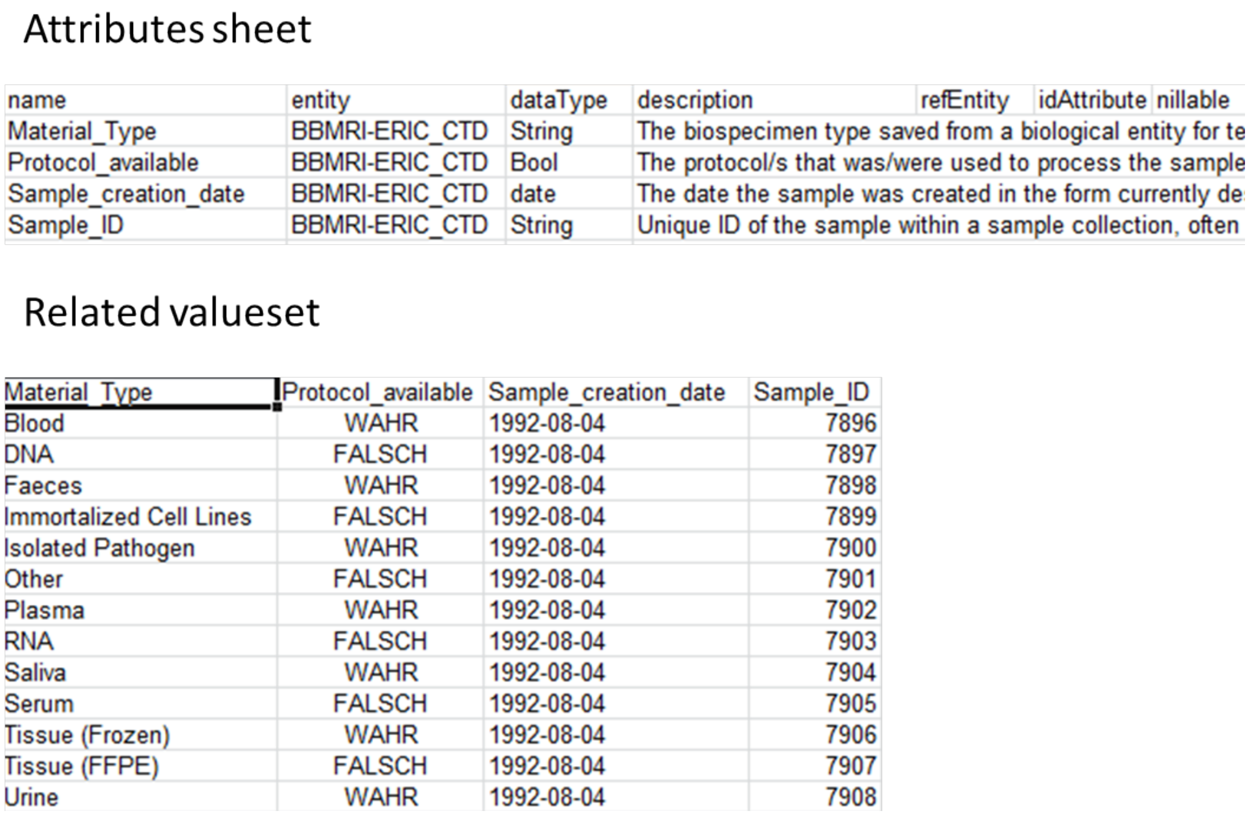


Figure 6. Hierarchical structure of an exemplary namespace.

The data elements are saved in a list of all data elements. The subgroups one by one repeat starting of getting their appropriate members, same here: data elements are saved in the list and the subgroups repeat starting. The previous illustration might clarify how the programme operates. The numbers of the different data elements and subgroups demonstrate the sequence in which the programme realises the hierarchical structure. For this reason, the numbers of the left side are the minor ones of the hierarchical tree. This shows that the programme will finish the left big "tree" first. The second one on the right hand will be processed after that.

Once that has been done, one data element after another from the list is picked out and stored in the EMX file. This file is an Excel file which consists of two different sheets. The first one is for the attributes; every column stands for another attribute and every row stands for another data element. The second sheet is a value set, where one can express permitted values. A small selection of the BBMRI-ERIC core schema and its related value set can be seen in Figure 7. The correctly written attributes, elements and permitted datatypes are necessary for the following steps of integrating data.

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 676550.



Attributes sheet

name	entity	dataType	description	refEntity	idAttribute	nillable
Material_Type	BBMRI-ERIC_CTD	String	The biospecimen type saved from a biological entity for te			
Protocol_available	BBMRI-ERIC_CTD	Bool	The protocol/s that was/were used to process the sample			
Sample_creation_date	BBMRI-ERIC_CTD	date	The date the sample was created in the form currently de			
Sample_ID	BBMRI-ERIC_CTD	String	Unique ID of the sample within a sample collection, often			

Related valueset

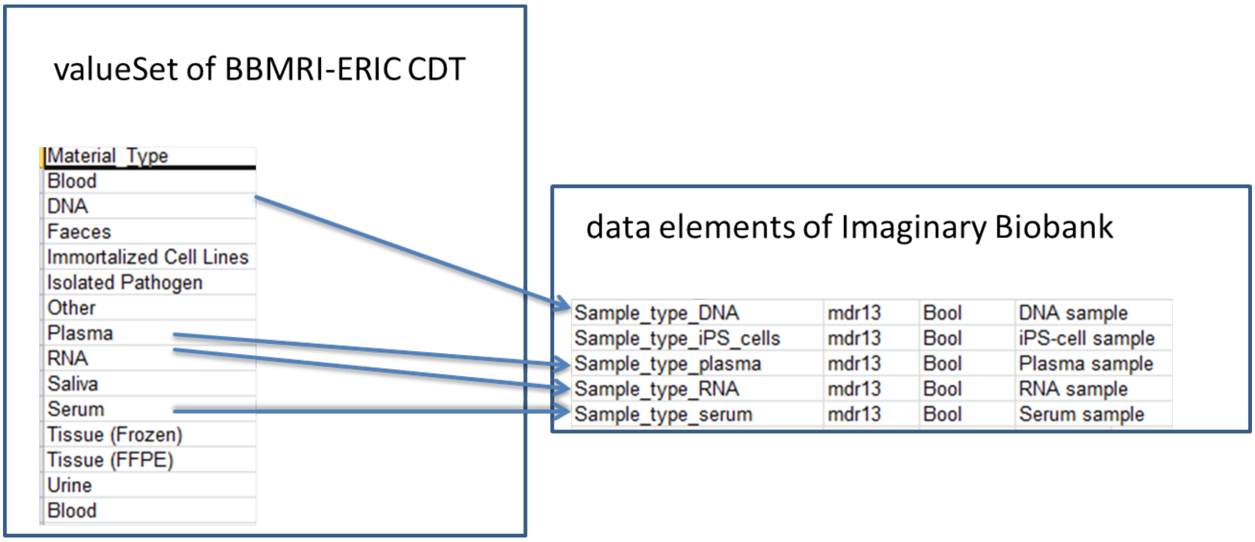
Material_Type	Protocol_available	Sample_creation_date	Sample_ID
Blood	WAHR	1992-08-04	7896
DNA	FALSCH	1992-08-04	7897
Faeces	WAHR	1992-08-04	7898
Immortalized Cell Lines	FALSCH	1992-08-04	7899
Isolated Pathogen	WAHR	1992-08-04	7900
Other	FALSCH	1992-08-04	7901
Plasma	WAHR	1992-08-04	7902
RNA	FALSCH	1992-08-04	7903
Saliva	WAHR	1992-08-04	7904
Serum	FALSCH	1992-08-04	7905
Tissue (Frozen)	WAHR	1992-08-04	7906
Tissue (FFPE)	FALSCH	1992-08-04	7907
Urine	WAHR	1992-08-04	7908

Figure 7. Small section of an attribute sheet and a related value set of the BBMRI-ERIC CTD namespace.

Once all steps are completed and there are no further un-processed data elements or subgroups in it, the file can be downloaded. Early experiences with MDR2MOLGENIS indicate that it is a very useful tool for in integrating data sets with MOLGENIS.

7. Open Challenges and Outlook

Our work on MOLGENIS is still in progress. Until now, details of MOLGENIS' internals still need further evaluation. Mapping the already described biobank data sets to the core schema of the BBMR-ERIC CTD schema, however, has been successful. There are different types of problems, which could not be solved until now. For example the "Imaginary Biobank" has its “Sample Type” described as a Boolean. Therefore, it contains different data elements as can be seen in Figure 8. The core schema only describes its “Material Type”, which is obviously the same, in an enumerated list of values. Figure 8 shows that the "Imaginary Biobank" has a similar data set in different datatypes. At this point, it remains to be analysed how to match those values.



valueSet of BBMRI-ERIC CDT

Material_Type
Blood
DNA
Faeces
Immortalized Cell Lines
Isolated Pathogen
Other
Plasma
RNA
Saliva
Serum
Tissue (Frozen)
Tissue (FFPE)
Urine
Blood

data elements of Imaginary Biobank

Sample_type_DNA	mdr13	Bool	DNA sample
Sample_type_iPS_cells	mdr13	Bool	iPS-cell sample
Sample_type_plasma	mdr13	Bool	Plasma sample
Sample_type_RNA	mdr13	Bool	RNA sample
Sample_type_serum	mdr13	Bool	Serum sample

Figure 8. Challenge of mapping different datatypes.

The integration of MOLGENIS into our harmonisation pipeline will be further tested and evaluated with other data sets. The creation of such testing material that is more similar to the core schema is important for understanding how MOLGENIS creates the different mapping expressions, what the critical parameters for a self-sufficient mapping are and how MOLGENIS classifies its own mapping expression. Also, it has to be further evaluated how MOLGENIS utilises its matching algorithms. In the testing phase some matches were automatically created, but some were only suggested. It is also possible to reuse already created matching expression due to another element where MOLGENIS did not suggest any. For the future it is essential to figure out, how to implement further mapping expressions in MOLGENIS.

Also MOLGENIS itself provides many different modules for mapping data sets. These tools have to be reviewed and it needs to be clarified in which cases they are useful (e.g. the “Mapping Service”, “SORTA”, “Tag Wizard” and “Ontology Manager”). It is important to clarify in which way these tools can be helpful for a better use of MOLGENIS.

As already mentioned, mapping with MOLGENIS is not as easy as it seems to be. Until now many questions arose while working with mapping expressions. There are many different mapping expressions needed in the harmonisation process. A person without any IT background may not be able to deal with these. This is the reason why it should be discussed if it is possible to develop something like a dictionnary or a repository for these mapping expressions and the corresponding meaning. This offers the possibility that every biobanker can upload data on his own. This may lead to personnel savings and it would make the process of mapping data sets more comfortable.

Also it has to be discussed, if a semi-automated translation service should be included in MDR2MOLGENIS. The translation services might be very helpful as the step before mapping the data. This step can facilitate the mapping process. Therefore, some data elements can be brought into the right schema without using MOLGENIS. It must be considered which languages might be good in supporting the mapping process.

This project has received funding from the *European Union’s Horizon 2020 research and innovation programme* under grant agreement No 676550.

8. Conclusion

According to our current state of knowledge, we recommend to use MOLGENIS as core software component for the harmonisation toolset. In addition to MOLGENIS, BiobankUniverse – an offspring of MOLGENIS –, MDR2MOLGENIS and further proprietary developments should be included in the mapping service. This approach and the recommended workflow can be seen in the following component/flow diagram.

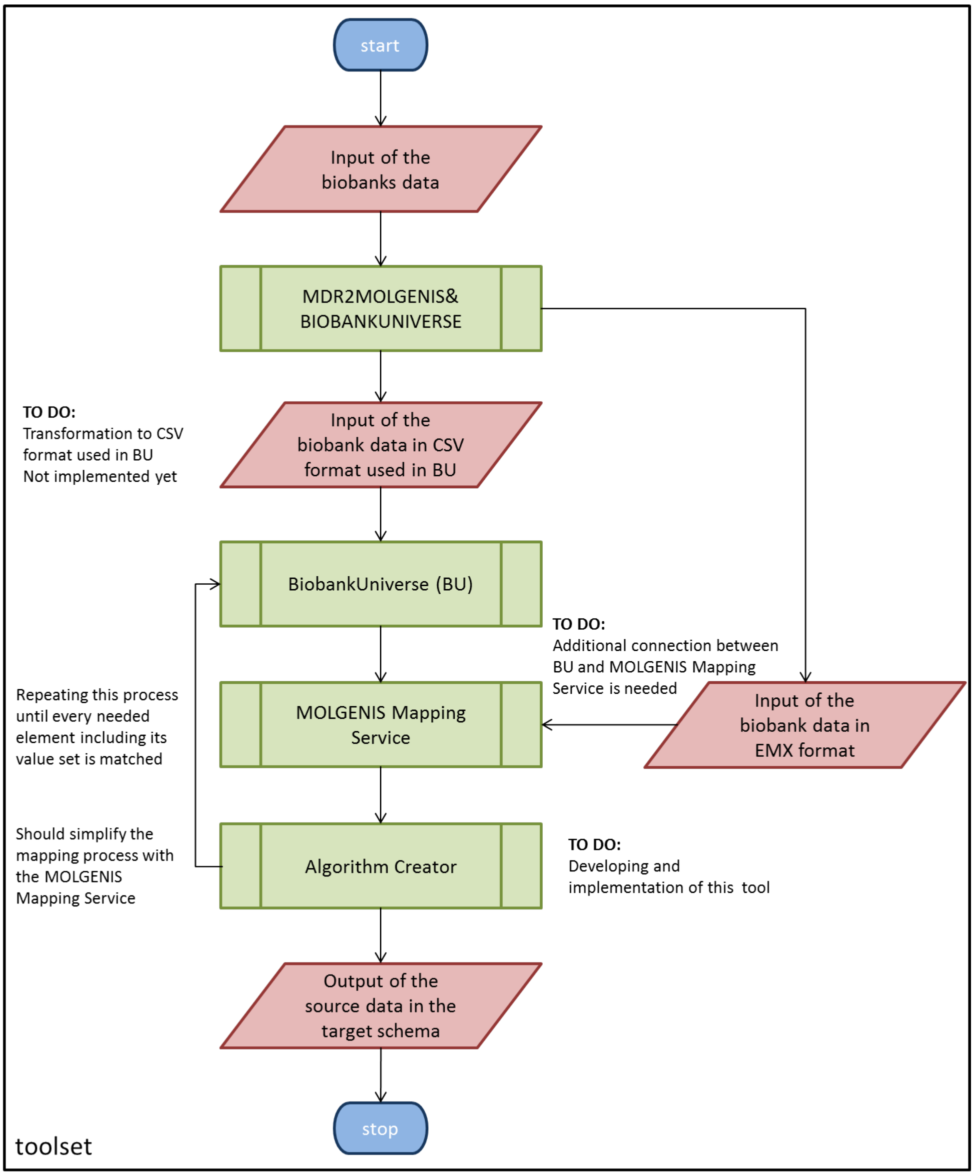

start
Input of the biobanks data
MDR2MOLGENIS& BIOBANKUNIVERSE
TO DO: Transformation to CSV format used in BU Not implemented yet
Input of the biobank data in CSV format used in BU
BiobankUniverse (BU)
TO DO: Additional connection between BU and MOLGENIS Mapping Service is needed
Repeating this process until every needed element including its value set is matched
MOLGENIS Mapping Service
Input of the biobank data in EMX format
Should simplify the mapping process with the MOLGENIS Mapping Service
Algorithm Creator
TO DO: Developing and implementation of this tool
Output of the source data in the target schema
stop
toolset

Figure 9 - Component and flow diagram of the toolset

A brief explanation should clarify the diagram below. The different tools are shown in green, the input and output data in red. The comments show the “TO DO’s”, which aren’t implemented yet. The toolsets workflow should start with the MDR2MOLGENIS and an additional extension of it, which also

This project has received funding from the *European Union’s Horizon 2020 research and innovation programme* under grant agreement No 676550.

converts the data into the CSV format needed in BiobankUniverse. The steps to perform the tasks of both tools are separated; therefore a connection between those must be established. One option would be that the toolset automatically opens the Biobank Universe overview and starts to automatically apply the matching algorithms and the improved GUI. For further editing the user can then switch to the MOLGENIS Mapping Service editing page. This part of the MOLGENIS Mapping Service provides the best features of MOLGENIS/connect. Moreover, the editing page of the MOLGENIS Mapping Service includes a feature to manually search the elements for a possible matching. By connecting the editing page, the opportunity to map the value sets is created. To facilitate this process, another tool must be implemented. This tool should create the algorithms, which are used for mapping the value set. The user then has to insert the target and source datatype as well as the related values. After submitting that information, this tool should create a JavaScript based algorithm, which can be inserted into the algorithm window of the MOLGENIS Mapping Service. This facilitates to map the whole data, with a very low effort for the biobanks.

The development of the new components for this recommended workflow shall be developed/implemented in the next 6 months of the ADOPT funding.

9. Medium/Long term BBMRI ERIC recommendation

In parallel to the above described work the future development roadmap and support structure of the MOLGENIS tools and BiobankUniverse as well as the joint roadmap to reintegrate both development strings again shall be clarified with the Groningen Bioinformatics Center (as the original developers of such tools). Depending on the Groningen Bioinformatics Center development/support roadmap for such tools modifications to the above recommended toolset might be required for the medium and long term perspective.

References

Ahlbrandt J, Brammen D, Majeed RW, *et al.* Balancing the Need for Big Data and Patient Data Privacy - An IT Infrastructure for a Decentralized Emergency Care Research Database. *Stud Health Technol Inform* 2014;205:750–4. doi:10.3233/978-1-61499-432-9-750

Ganslandt T, Mate S, Helbing K, *et al.* Unlocking Data for Clinical Research – The German i2b2 Experience. *ACI* 2011;2:116–27. doi:10.4338/ACI-2010-09-CR-0051

Kadioglu D, Weingardt P, Ückert F, Wagner, T. Samply.MDR – Ein Open-Source-Metadaten-Repository. HEC 2016: Health — Exploring Complexity 2016 Joint Conference of GMDS, DGEpi, IEA-EEF, EFMI. German Medical Science, doi: 10.3205/16gmds149, 2016.

Lablans M, Kadioglu D, Mate S, *et al.* Strategies for Biobank Networks. *Bundesgesundheitsbl* Published Online First: 11 January 2016. doi:10.1007/s00103-015-2299-y

Lablans M, Kadioglu D, Muscholl M, *et al.* Exploiting Distributed, Heterogeneous and Sensitive Data Stocks While Maintaining the Owner's Data Sovereignty. *Methods Inf Med* 2015;54. doi:10.3414/ME14-01-0137

Majeed RW, Röhrig R. Automated Realtime Data Import for the i2b2 Clinical Data Warehouse: Introducing the HL7 ETL Cell. *Stud Health Technol Inform* 2012;180:270–4.

Mate S, Kadioglu, Majeed RW, Stöhr, Folz M, Vormstein P, Storf H, Brucker DP, Keune D, Zerbe N, Hummel M, Senghas K, Prokosch H-U, Lablans M. Proof-of-Concept Integration of Heterogeneous Biobank IT Infrastructures into a Hybrid Biobanking Network. Submitted to GMDS 2017, 2017.

Mate S, Vormstein P, Kadiogu D, Majeed RW, Lablans M, Prokosch H-U, Storf H. On-The-Fly Query Translation Between i2b2 and Samply in the German Biobank Node (GBN) Prototypes. Submitted to GMDS 2017, 2017.

McMurry AJ, Murphy SN, MacFadden D, *et al.* SHRINE: Enabling Nationally Scalable Multi-Site Disease Studies. *PLoS ONE* 2013;8:e55811. doi:10.1371/journal.pone.0055811

Murphy SN, Weber GM, Mendis ME, *et al.* Serving the Enterprise and Beyond with Informatics for Integrating Biology and the Bedside (i2b2). *J Am Med Inform Assoc* 2010;17:124–30.

Pang C, van Enckevort D, de Haan M, Kelpin F, Jetten J, Hendriksen D, de Boer T, Charbon B, Winder E, van der Velde KJ, Doiron D, Fortier I, Hillege H, Swertz MA. MOLGENIS/connect: a system for semi-automatic integration of heterogeneous phenotype data with applications in biobanks. Bioinformatics. 2016 Jul 15;32(14):2176-83.

Pang C, Hendriksen D, Dijkstra M, van der Velde KJ, Kuiper J, Hillege HL, Swertz MA. BiobankConnect: software to rapidly connect data elements for pooled analysis across biobanks using ontological and lexical indexing. J Am Med Inform Assoc. 2015a Jan;22(1):65-75.

Pang C, Sollie A, Sijtsma A, Hendriksen D, Charbon B, de Haan M, de Boer T, Kelpin F, Jetten J, van der Velde JK, Smidt N, Sijmons R, Hillege H, Swertz MA. SORTA: a system for ontology-based re-coding and technical annotation of biomedical phenotype data. Database (Oxford). 2015b Sep 18;2015. pii: bav089. doi: 10.1093/database/bav089.

Storf H, Schaaf J, Kadioglu D, Göbel J, Wagner TO, Ückert F. [Registries for rare diseases : OSSE - An open-source framework for technical implementation]. Bundesgesundheitsblatt Gesundheitsforschung Gesundheitsschutz. 2017 Mar 13. doi: 10.1007/s00103-017-2536-7.

Swertz MA, Dijkstra M, Adamusiak T, van der Velde JK, Kanterakis A, Roos ET, Lops J, Thorisson GA, Arends D, Byelas G, Muilu J, Brookes AJ, de Brock EO, Jansen RC, Parkinson H. The MOLGENIS toolkit: rapid prototyping of biosoftware at the push of a button. BMC Bioinformatics. 2010 Dec 21;11 Suppl 12:S12. doi: 10.1186/1471-2105-11-S12-S12.

van Ommen G-JB, Törnwall O, Bréchot C, *et al.* BBMRI-ERIC as a Resource for Pharmaceutical and Life Science Industries: The Development of Biobank-Based Expert Centres. *European Journal of Human Genetics* 2015;23:893–900. doi:10.1038/ejhg.2014.235

ADOPT BBMRI-ERIC

Grant Agreement no. 676550

DELIVERABLE REPORT

Deliverable no	D3.5
Deliverable Title	Provision of an ETL Service for Pan-European Data Mapping & Harmonization - Deployment of the Final Toolset
Contractual delivery month	M30 (March 2018)
Responsible Partner	BBMRI-ERIC CS IT WP8
Author(s)	Sebastian Mate[1], Christian Knell[1], Christina Schüttler[1], Niina Eklund[2], Kaisa Silander[2], Salla-Maaria Pätsi[2], Petr Holub[3], Hans-Ulrich Prokosch[1] *[1] Medical Informatics, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU), Erlangen, Germany* *[2] Genomics and Biomarkers Unit, National Institute for Health and Welfare, Helsinki, Finland* *[3] BBMRI-ERIC, Graz, Austria*
Actual delivery date	2018-10-02

Provision of an ETL Service for Pan-European Data Mapping & Harmonization – Deployment of the Final Toolset

Executive Summary

BBMRI-ERIC is a pan-European consortium with the goal to establish and operate a research infrastructure to facilitate the access to the sample collections of participating biobanks (van Ommen et al., 2015). ADOPT BBMRI-ERIC aims to support the implementation of the BBMRI-ERIC research infrastructure by providing the necessary IT services in cooperation with partners which are funded by the BBMRI ERIC Common Service (CS) IT grants.

This document continues the series of reports regarding the deployment of a toolset for the mapping of the distinct biobanking terminologies, and its associated data integration. This deliverable (BBMRI-ERIC CS-IT D8.4 / ADOPT 3.5 M30) describes the functionalities and the development of a final toolset prototype, which is based on enhancements of previously developed tools (e.g. the Samply Metadata Repository (Kadioglu, Weingardt, Ückert, & Wagner, 2016; Lablans, Kadioglu, Muscholl, & Ückert, 2015; Storf et al., 2017)) and new components integrated into a

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

data harmonization pipeline. This document further describes the currently proposed workflow for applying those tools to integrate a biobank (and its respective local data sources/data elements) in the BBMRI-ERIC network. Finally, the document also describes the evaluation results from applying the final Ontology-Based Toolset for Mapping of the Biobanking Terminologies to data provided by ten European biobanks within BBMRI ERIC ADOPT WP 3.

Content

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

1 Introduction

This deliverable covers one of the specific objectives of ADOPT BBMRI-ERIC (hereinafter referred to as “ADOPT”) WP3 Task 3, namely the development and deployment of a toolbox to support the mapping of data attributes/value lists, biobank metadata, as well as clinical case/sample description attributes to a central core terminology. This also includes the derivation of ETL processes based on these mappings.

The necessity of such a mapping toolset is due to the heterogeneity of data elements from different sources. Every partnering institution usually refers to a particular local data scheme adjusted for its own use. This proceeding might create data collections that are insufficiently described and consequently unsuitable for secondary use, e.g. future research performed by external researchers. In order to avoid this pitfall and to ensure the interoperability between originally heterogeneous data sources, a toolbox has been developed. This toolbox provides the software components to support the mapping of the biobanks’ existing data on an agreed upon common terminology for the purpose of data harmonization.

Aiming to kick-start BBMRI-ERIC's data collection and harmonization, ADOPT decided to collect colorectal cancer data as the first fully implemented disease unit. This disease is one of the most common cancers, and most national nodes already had colorectal cancer programs to work with. The aim was to make the collected and validated data and samples available in the BBMRI-ERIC portal. This data collection is commonly known as the *Colon Cancer Data Collection* (CCDC) within ADOPT. The toolbox described here has been utilized for loading data from different European Biobanks into the CCDC.

In the following, the report outlines the specifications and functionalities that are considered essential for the mapping/harmonization process. Moreover, the development of the final toolset will be described.

2 Previous Work

Earlier developments (e.g. the Samply MDR (Kadioglu et al., 2016), Samply.EDC.OSSE (Storf et al., 2017) and concepts for “on-the-fly query translations” (Mate, Vormstein, et al., 2017b), as well as the federated architecture of data repositories based on local data warehouse implementations (Mate, Kadioglu, et al., 2017a), *Connector* and a *Locator* tool (Proynova et al., 2017)) on which our current new work is based, have come from the German DKTK project (Steffens et al., 2012) and early prototypes developed in the German Biobank Node (GBN) project (Mate, Kadioglu, et al., 2017a). The BBMRI-ERIC *Negotiator* has been developed in order to facilitate the access procedure where the requesters actually request access to the potentially available samples and data sets. Details of these concepts had already been described in ADOPT Delivera-

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

ble 3.5 *Ontology-Based Toolset for Mapping of the Biobanking Terminologies - Provision of a Terminology Service for Semantic Ontology Mapping – First Toolset Prototype* as well as ADOPT Deliverable 3.6 *Biobank Connector Specification and Reference Connector Implementation*.

While ADOPT deliverable 3.6 in general foresees a federated design with local DWH implementations at the site of each participating biobank, for the CCDC establishment in ADOPT the decision was made to implement a central colon cancer database (compare D 3.2, *User Interface for Collection of the Colon Cancer Cases and Database*).

For providing data for the CCDC database two options were offered to the BBMRI-ERIC biobanks:

- To enter data manually via the CCDC graphical user interface (based on the Samply.EDC.OSSE), and
- to load data via an ETL process providing an XML import file, corresponding to a given XSD-file.

Since the data store development for the distributed local DWH implementations at the end of 2017 was not finalized yet, and the BBMRI-ERIC biobanks, which aimed to deliver their colon cancer data for the CCDC in a semi-automated manner and required support for their ETL and data harmonization processes, the decision was made to align the specification and development of the ADOPT data harmonization toolset to the CCDC specification.

Further, it was decided to define one core data terminology for the CCDC and to perform the data harmonization as an integrated part of the ETL processes (instead of an “on-the-fly” query mapping as applied in the GBN prototype). In other words, the biobanks have to perform the data harmonization/integration locally, and the connector for the CCDC database will store only already harmonized data. Thus, the data harmonization toolset specified and developed within ADOPT WP 3 Task 3 was specifically designed to support the BBMRI-ERIC biobanks’ ETL and data harmonization process for loading data into the CCDC database.

It is envisioned that similar ETL processes can then be implemented by the biobanks in order to feed their instances of generic Connectors when connecting to the Locator service; the major difference will be a slightly different specific data structure and specification of the XML structure.

3 The ADOPT Data Harmonization Toolset

Data harmonization is a complicated process that needs to be supported by software tools while utilizing standardized metadata structures as means for describing the data in a standardized,

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

computer-readable fashion. For this purpose, an ADOPT data harmonization toolset has been developed, which comprises the following tools:

- Samply.MDR - for the semantically precise definition of data elements,
- MDRExtractor - to make the definitions of the MDR usable in the ETL tools,
- TablePreprocessor - to integrate tabular data (e.g. Excel) with the ETL approach,
- MDRMatcher - to automatically create mappings between data elements,
- Mapping GUI - to visualize and manually correct the automatic mappings,
- ETLHelper - to transform the actual data according to these mappings.

All of these tools are Open Source and available at GitHub, and complemented with a document containing user instructions (currently located at: https://github.com/sebmate/ADOPT-BBMRI-ERIC-ETL-Tools/).

In the following sections, those tools are described and illustrated as components being stepwise applied within the ETL pipeline for loading colon cancer sample data into the CCDC database.

4 The ETL / Data Harmonization Workflow

4.1 Final Architecture Design

Figure 1 outlines the ETL process / data flow which needs to be pursued within BBMRI-ERIC CS-IT by each biobank to load data semi-automatically into the CCDC connector as the target system. The CCDC connector currently only supports data imports via an XML formatted file (lower right corner of Figure 1). Thus, a biobank needs to create such an XML file matching the corresponding XSD-file scheme. Since

1. the biobank's data elements and their values in general do not directly match the core data terminology for the CCDC and
2. the biobanks mostly were not able to provide the required XML file format,

a simple, but generic CSV format (based on the very flexible Entity-Attribute-Value (EAV)-concept described in (Nadkarni et al., 1999)) was defined (see chapter 4.2.1) as the format in which the biobanks, in order to use the BBMRI-ERIC infrastructure, should provide their data.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

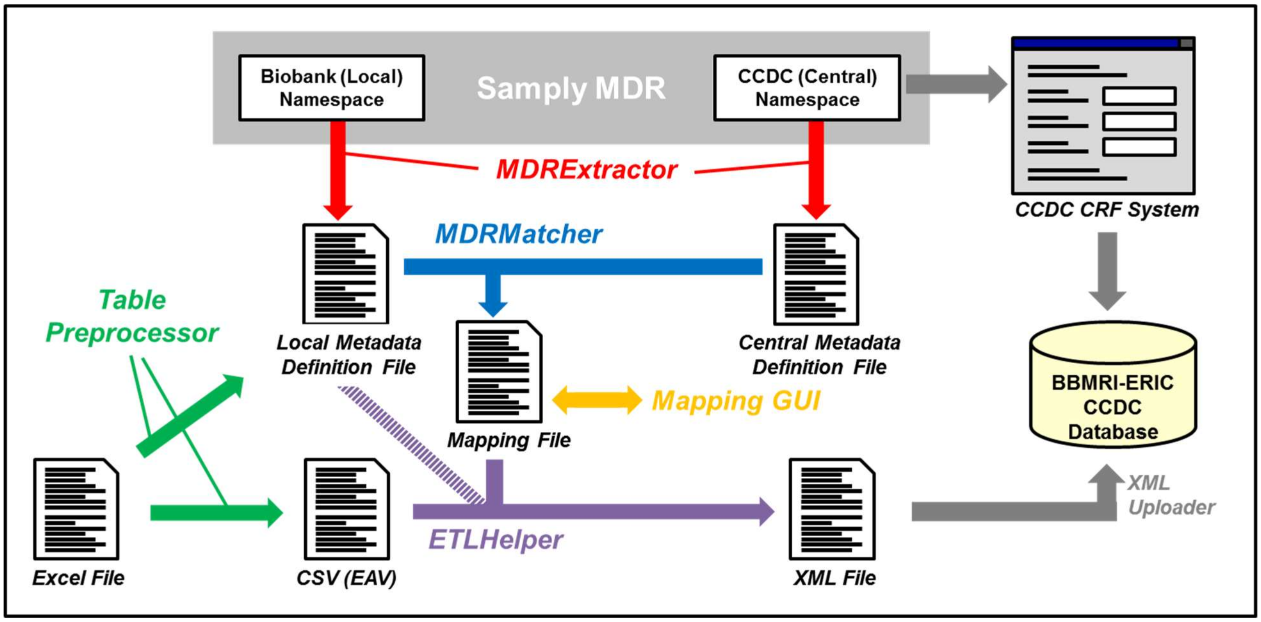


Figure 1: Data flow in the ADOPT ETL pipeline.

It was expected that the biobanks are able to extract their sample/donor data from their local primary systems and provide it in such a flexible CSV format. The EAV-structure was chosen, because of its flexible capability to provide multiple values (e.g. different measurement results of a laboratory value at different time points in a patient history) for single data elements (even if the maximum number of value instances is not known in advance).

Within the support processes for the biobanks, however, it became clear that even this CSV/EAV format was too complicated for the biobanks and that the only format they could provide, would be a flat excel file, were multiple values for one data element would be entered in one Excel cell separated by a predefined separator (semicolon “;” was chosen). Thus, it was decided to enhance the data harmonization toolkit with the **TablePreprocessor** tool, which would parse such an Excel file and create the corresponding CSV/EAV-file. The fact that this does lead to an information loss (because the timestamps for dedicated values cannot be provided in this format) was accepted, since the current specification of the CCDC database also does not support timestamps for multiple value instances for one data element.

As the **MDRMatcher’s** name implies, the idea behind the tool is to lexically match the contents of MDR namespaces. Therefore, it was another original prerequisite for each biobank to concisely describe the metadata of their local data elements in the biobank's specific (local) MDR namespace.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

In the practical efforts to support ten biobanks (as of July 2018) in this harmonization process in CCDC pilot, however, it also became clear that it was obviously too complicated for the local biobank staff to describe their local data elements in the MDR precisely and concisely, although the procedure for this was fully documented by BBMRI-ERIC. For example, data element names or values provided in the actual data had a slightly different spelling compared to the corresponding MDR entries (for example, upper/lower case, spaces or hyphens between multiple word names). Since the process of iteratively correcting such inconsistencies with the biobanks became too time-consuming to meet the rather tight CCDC deadlines, it was decided to accept a small loss of information by not using the metadata from the MDR but extracting it directly from the Excel files.

This alternative extraction of metadata (from the Excel file) has been integrated into the **TablePreprocessor**. The tool has been extended to automatically derive a "local" metadata definition file for the biobank (see Figure 1) while processing and converting the Excel file into the CSV/EAV format. This metadata definition, because it was extracted electronically, complies exactly with the data from the biobank's colon cancer Excel file.

This extraction of metadata from the Excel file has the disadvantage (compared to the originally planned use of the MDR) of leading to a minimal loss of information. First, of course no metadata for unused data elements or value sets can be extracted from an Excel file. For example, a prostate cancer data file will only contain male patients and therefore result in an incomplete value set definition for the "sex" data element. In the context of the ETL result, however, it makes no difference whether mappings are already created for such potentially occurring value sets. Since there is no data to be integrated for these metadata anyway, the absence of mappings has no impact on the final integrated data. Second, there is no hierarchical grouping information in a flat Excel table (unlike in the MDR, where one can create such information). As we will describe later, such hierarchical grouping knowledge can potentially lead to better results from the matching algorithm. In practice (see chapter Results), however, it has been shown that the automatic mapping results are respectable even without this hierarchy information.

Thus, the original prerequisites for using the data harmonization toolkit to

- provide the colon cancer data in CSV/EAV format
- while describing the original source file metadata in the Samply.MDR (Biobank Namespace)

was lowered, so that the approach can now accept

- the provision of data in flat Excel files
- without any manual MDR metadata entries by the biobank.

The data tool **TablePreprocessor** now fulfills the task to

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

- create the CSV/EAV format from the flat Excel file; and also
- create the respective metadata entries compatible to those from the Samply.MDR.

In Figure 1 the data harmonization toolkit's components and their application within the different steps of the ETL (data harmonization) process are indicated by arrows in different colors:

- The **TablePreprocessor** tool (green) parses the *Excel file* (or any other tabular data) and rotates the data into the CSV/EAV format. It also generates a *Local Metadata Definition File*.
- If the data is provided in the CSV/EAV format and the biobank has entered its metadata into the Biobank Namespace of the MDR, then the **MDRExtractor** tool (red) can be used to create this *Local Metadata Definition File.* Similarly, a *Central Metadata Definition File* is created by the **MDRExtractor** from the MDR namespace, which contains the CCDC core terminology.
- Both Metadata Definition Files can be matched with the **MDRMatcher** tool (blue). This is described in detail in chapter 4.3. This tool generates a *Mapping File*, which has to be verified by a human biobank expert.
- The *Mapping File* can be manually edited with the **Mapping GUI** program (yellow). This is required to verify the proposed mappings from the MDRMatcher tool. The tool has recently been extended to generate statistics about the mapping quality (by comparing the software-generated with the user-curated mappings), which will be outlined later in this document.
- Finally, the **ETLHelper** tool (purple) can process the *CSV/EAV file* in combination with the *Mapping File* and metadata definitions to transform the data into the final XML representation. This XML file can then be uploaded into the CCDC system.

The MDR plays a central role in this concept. It is used to precisely describe the data elements, both in the primary source data of the biobanks and in the target data set. It consists of a large number of different sections to describe the data of the target data set and of each biobank. These sections are called "namespaces". The idea is that (when using the CSV/EAV approach) each biobank clearly describes the data structures and formats of its own primary data sources in its own namespace in the MDR. This is usually done manually, using the corresponding graphical user interface. However, it may also be possible (depending on the possibilities of the source systems) to automatically extract/generate this metadata from a biobank's source system. Then such metadata have to be transformed into the MDR import format in order to load them into the MDR with an automatic importer. Further, as explained above, such metadata definitions can also be generated based on the Excel data file provided by the biobank, by the **TablePreprocessor** tool.

When called, the **MDRExtractor** can be parameterized via a configuration file (see the documentation of the ETL tools), so that it extracts the contents of a MDR namespace (which is associated with one biobank) into a metadata file, which can later then be used by the **MDRMatcher**.

This **MDRMatcher** compares the contents of two such metadata files in order to find mappable data elements and attributes of value lists (see 4.3 for details). Since a fully automatic mapping is practically impossible if we also require reliability and accuracy, the **MDRMatcher** creates an output file (mapping file) as a result of a match run, in which potentially mappable data elements are placed next to each other and provided with a corresponding similarity score. It is important to note, that the term "matching" is used in context of generating mapping proposals automatically. When the match suggestions are approved, they are confirmed by a human expert as "mappings".

The task of a human expert (the person processing/importing the biobank data) is then to review these proposals with the **Mapping GUI** and, if necessary, to correct them. The result of such a semi-automatic matching process is a mapping file, which is then used to create a "set of rules" for the mapping process of the biobank's data to the target format of the local data warehouse in the final step of the transformation process.

This final step, the actual ETL process, is based on the fact that the selected set of biobank data (which is to be made available for the sample requests in the local data warehouse) must first be extracted from the site-specific local source systems and exported to the predefined *Excel* or *CSV/EAV file* format. This *CSV/EAV file* and the mapping file previously created for the respective biobank serve as inputs for the transformation process, which is implemented using the **ETLHelper** tool. The **ETLHelper** tool uses the internally generated "set of rules" based on the entries in the mapping file to convert the data from the CSV file into the XML structure corresponding to the XSD scheme defined for the CCDC or a Connector database.

For the CCDC system, an uploader is provided by BBMRI-ERIC CS IT WP2 for the final loading step, which imports the data in the XML structure into the database of the CCDC data warehouse.

4.2 Data Input Formats for the TablePreprocessor

The CCDC (and in the future the Connector) natively supports a special XML format for data input. This format has been specified and was implemented by BBMRI-ERIC CS IT WP2. After several development iterations, the ETL approach (as described in this document) allows two new input formats:

- The *CSV/EAV format*, which has been specified by ADOPT WP3 / CS-IT WP8 in the document "Mapping and ETL – V13.docx", and is briefly summarized in chapter 4.2.1

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

- The *Excel format*, which is described in chapter 4.2.2.

The following sections outline the two latter formats and the translation between them.

4.2.1 CSV/EAV Input Format (with MDR-utilization)

This format is similar to the generic Entity-Attribute-Value (EAV) format (which is described e.g. in (Nadkarni et al., 1999)), with the exception of the additional timestamp and instance columns, as shown in Table 1.

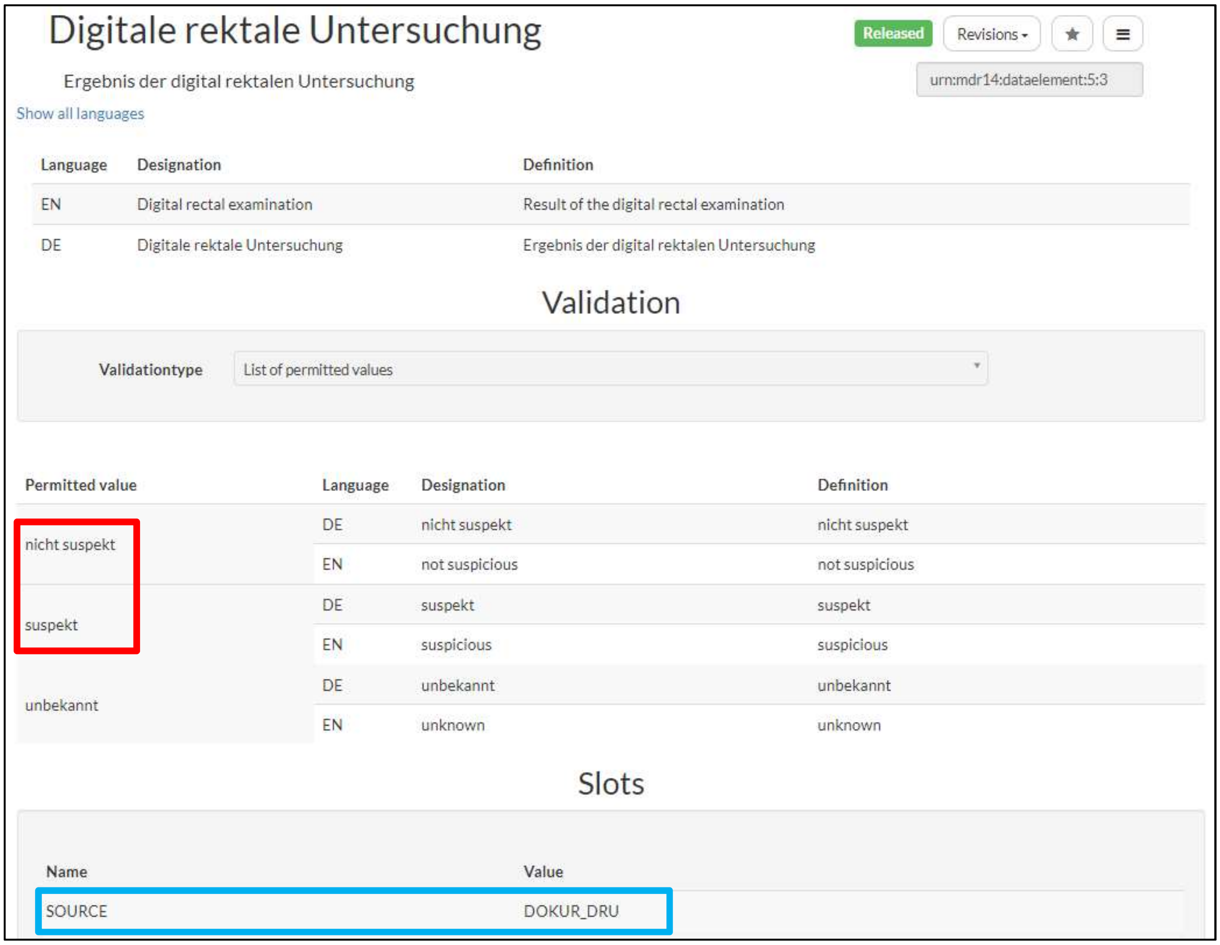
CASEID	CONCEPT	VALUE	TIMESTAMP	INSTANCE
100400001	DOKUR_TumLoc	links	2007-03-22 15:35:00	1
100123101	DOKUR_TumLoc	links	2013-08-15 08:30:00	1
100400001	DOKUR_TumLoc	rechts	2007-03-22 15:35:00	1
100023411	DOKUR_TumLoc	unbekannt	1999-10-14 20:04:00	1
100400001	DOKUR_DRU	nicht suspekt	2007-03-22 16:30:00	1
100023411	DOKUR_DRU	suspekt	1999-10-16 21:00:00	1
100023411	DOKUR_Sex	männlich	1999-10-16 21:00:00	1
102300001	DOKUR_Sex	männlich	2007-03-22 15:35:00	1
102345101	DOKUR_Sex	männlich	2013-08-15 08:30:00	1
123453457	DOKUR_Sex	männlich	2012-02-01 09:10:00	1
123589644	DOKUR_Sex	männlich	1999-10-14 20:04:00	1
100400001	DOKUR_TNM_T	2	2007-03-22 15:35:00	1
100400001	DOKUR_TNM_N	0	2007-03-22 15:35:00	1
100400001	DOKUR_TNM_M	0	2007-03-22 15:35:00	1
100400001	DOKUR_TNM_T	3	2007-08-12 18:58:00	2
100400001	DOKUR_TNM_N	2	2007-08-12 18:58:00	2
100400001	DOKUR_TNM_M	1	2007-08-12 18:58:00	2
…	…	…	…	…

Table 1: EAV input format (example). Example taken from the document "Mapping and ETL – V13.docx".

In contrast to "normal" relational tables, where each medical concept is stored in a distinct column while each patient's data is captured in a distinct row, the EAV format "rotates" this data by 90°. This format allows to easily define multiple instances of a data element while avoiding complex relational table schemas (note that in the above example e.g. the patient with ID 100400001 has two tumor locations, "TumLoc", one on the left (links) and one on the right (rechts)).

A very important aspect is the connection to the metadata (in the MDR). The entries in the "CONCEPT" column of the input file are referring to designators in a biobank's source system. Those entries are typically in the local language of the biobank (German in the example of Table

1). In order to support an automated data element names' and data values' matching to the CCDC core terminology, those entries in the MDR however need to be present also in English. Therefore the original language designator names, as they appear in the CONCEPT column, need to be specified in the respective "SOURCE" slot of the MDR (indicated with the blue boxes in Table 1 and Figure 2). Similarly, the entries in the VALUE column have to exactly match the permitted values in the original language (red boxes).



Digitale rektale Untersuchung

Ergebnis der digital rektalen Untersuchung

Language	Designation	Definition
EN	Digital rectal examination	Result of the digital rectal examination
DE	Digitale rektale Untersuchung	Ergebnis der digital rektalen Untersuchung

Validation

Validationtype: List of permitted values

Permitted value	Language	Designation	Definition
nicht suspekt	DE	nicht suspekt	nicht suspekt
	EN	not suspicious	not suspicious
suspekt	DE	suspekt	suspekt
	EN	suspicious	suspicious
unbekannt	DE	unbekannt	unbekannt
	EN	unknown	unknown

Slots

Name	Value
SOURCE	DOKUR_DRU

Figure 2: Metadata definition of CSV/EAV CONCEPT and VALUE entries in the MDR (example).

The fifth column of the CSV/EAV file, "INSTANCE", is used for the grouping of instances, which belong to the same clinical event type. In the above example, there are two instances of TNM values for the patient with ID 100400001. To allow for distinguishing between these two, different instance numbers have been associated with each of those entries. The import data

Horizon 2020

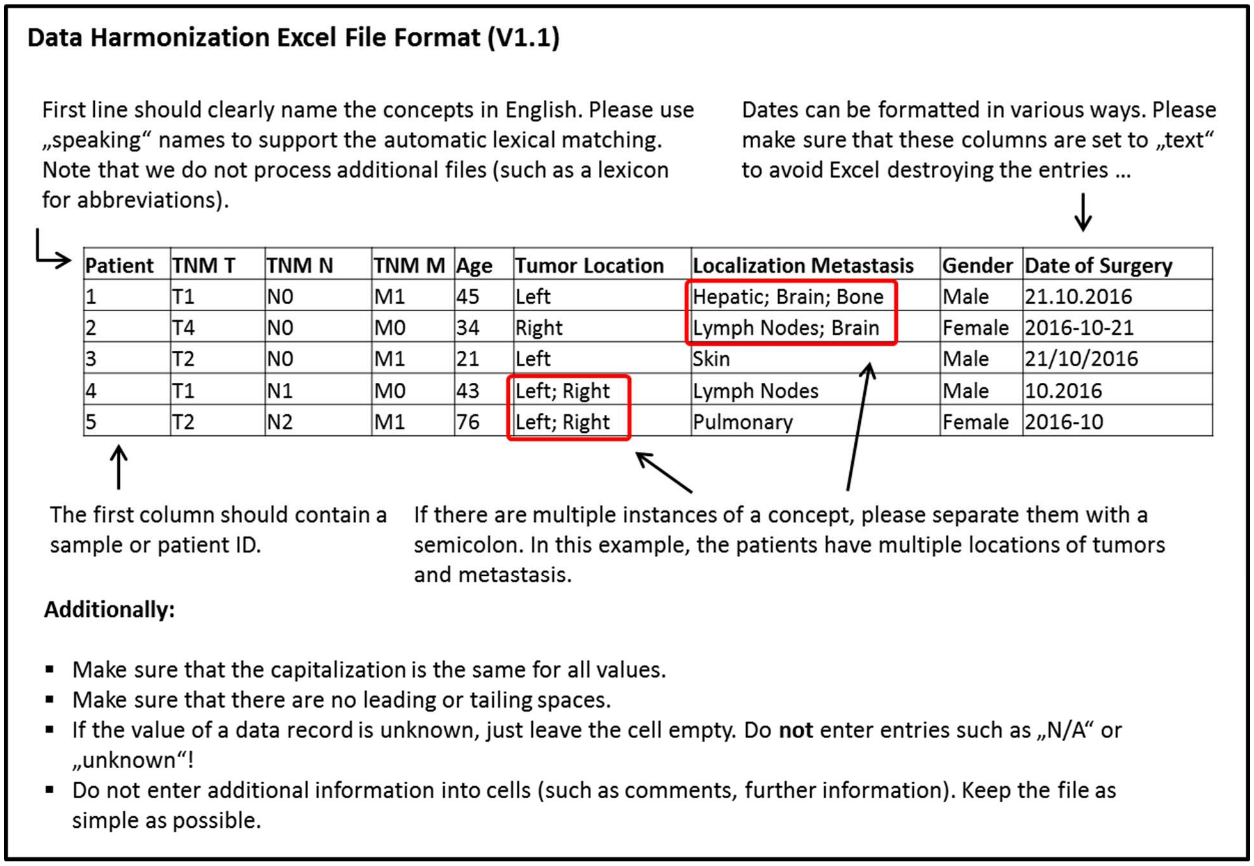
ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

should carry such an instance numbering, especially for follow-up-like data. It is also important for associating multiple values with e.g. multiple tumors.

Even though in some cases these instance numbers could be derived from the timestamp (in the example, the two instances carry different timestamps, but the timestamps are the same for each data element in each instance), since not all biobanks can provide timestamps, they are treated as optional. Because of this the "INSTANCE" column is mandatory and the CCDC data import relies only on the "INSTANCE" column for data grouping.

4.2.2 Excel Input Format (without MDR-utilization)

Since many biobanks were not able to transform their original source system data into the CSV/EAV file format an additional tabular flat Excel file format was defined. This format can deal with multiple instances by separating cell entries with semicolons (compare Figure 3). In contrast to the CSV/EAV format, the timestamp feature however is lost.



Data Harmonization Excel File Format (V1.1)

First line should clearly name the concepts in English. Please use „speaking" names to support the automatic lexical matching. Note that we do not process additional files (such as a lexicon for abbreviations).

Dates can be formatted in various ways. Please make sure that these columns are set to „text" to avoid Excel destroying the entries ...

Patient	TNM T	TNM N	TNM M	Age	Tumor Location	Localization Metastasis	Gender	Date of Surgery
1	T1	N0	M1	45	Left	Hepatic; Brain; Bone	Male	21.10.2016
2	T4	N0	M0	34	Right	Lymph Nodes; Brain	Female	2016-10-21
3	T2	N0	M1	21	Left	Skin	Male	21/10/2016
4	T1	N1	M0	43	Left; Right	Lymph Nodes	Male	10.2016
5	T2	N2	M1	76	Left; Right	Pulmonary	Female	2016-10

The first column should contain a sample or patient ID.

If there are multiple instances of a concept, please separate them with a semicolon. In this example, the patients have multiple locations of tumors and metastasis.

Additionally:

- Make sure that the capitalization is the same for all values.
- Make sure that there are no leading or tailing spaces.
- If the value of a data record is unknown, just leave the cell empty. Do **not** enter entries such as „N/A" or „unknown"!
- Do not enter additional information into cells (such as comments, further information). Keep the file as simple as possible.

Figure 3: Excel Spreadsheet input format specification as sent to the CCDC biobanks.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

4.2.3 Translation between Excel and CSV/EAV Format (TablePreprocessor)

Data provided in the Excel file format can automatically be translated into the CSV/EAV format (see Table 2 and Table 3).

Patient ID	C_1	C_2	…	C_n
P_1	$v_{1,1,1}$; $v_{1,1,2}$; … $v_{1,1,a}$	$v_{2,1,1}$; $v_{2,1,2}$; … $v_{2,1,d}$	…	$v_{n,1,1}$; $v_{n,1,2}$; … $v_{n,1,g}$
P_2	$v_{1,2,1}$; $v_{1,2,2}$; … $v_{1,2,b}$	$v_{2,2,1}$; $v_{2,2,2}$; … $v_{2,2,e}$	…	$v_{n,2,1}$; $v_{n,2,2}$; … $v_{n,2,h}$
…	…	…	…	…
P_m	$v_{1,m,1}$; $v_{1,m,2}$; … $v_{1,m,c}$	$v_{2,m,1}$; $v_{2,m,2}$; … $v_{2,m,f}$	…	$v_{n,m,1}$; $v_{n,m,2}$; … $v_{n,m,i}$

Table 2: Excel file format.

Patient ID	Concept	Value	Instance
P_1	C_1	$v_{1,1,1}$	1
P_1	C_1	$v_{1,1,2}$	2
P_1	C_1	…	…
P_1	C_1	$v_{1,1,a}$	a
P_2	C_1	$v_{1,2,1}$	1
P_2	C_1	$v_{1,2,2}$	2
P_2	C_1	…	…
P_2	C_1	$v_{1,2,b}$	b
…	C_1	…	…
P_m	C_1	$v_{1,m,1}$	1
P_m	C_1	$v_{1,m,2}$	2
P_m	C_1	…	…
P_m	C_1	$v_{1,m,c}$	c
…	…	…	…
P_m	C_n	$v_{n,m,1}$	1
P_m	C_n	$v_{n,m,2}$	2

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

P_m	C_n	…	…
P_m	C_n	$v_{n,m,i}$	i

Table 3: Corresponding CSV/EAV file format.

In both formats, a number of n medical concepts (C_1, C_2, ... C_n) is associated with m patients (P_1, P_2, ... P_m). Each of these tuples can have a vector of x values, $v_{patient,concept,1}$, $v_{patient,concept,2}$, ... $v_{patient,concept,x}$ or can be empty. For example, in the Figures 4 and 5, the vector of values associated with patient 2 and concept 1 is highlighted in pale blue and comprises the values $v_{1,2,1}$, $v_{1,2,2}$ ... $v_{1,2,b}$ (with the length of the vector $x = b$ in this case).

As an example, consider the following Excel file shown in Table 4.

Patient ID	Age	Tumor Location	Localization Metastasis
1	45	Left	Hepatic; Brain; Bone
2	34	Right	Lymph Nodes; Brain
3	21	Left	Skin
4	43	Left; Right	Lymph Nodes
5	76	Left; Right	Pulmonary

Table 4: Example data in Excel format.

After the transformation into the CSV/EAV format, the data is as shown in Table 5.

Patient ID	Concept	Value	Instance
1	Age	45	1
1	Tumor Location	Left	1
1	Localization Metastasis	Hepatic	1
1	Localization Metastasis	Brain	2
1	Localization Metastasis	Bone	3
2	Age	34	1
2	Tumor Location	Right	1
2	Localization Metastasis	Lymph Nodes	1
2	Localization Metastasis	Brain	2
3	Age	21	1
3	Tumor Location	Left	1

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

3	Localization Metastasis	Skin	1
4	Age	43	1
4	Tumor Location	Left	1
4	Tumor Location	Right	2
4	Localization Metastasis	Lymph Nodes	1
5	Age	76	1
5	Tumor Location	Left	1
5	Tumor Location	Right	2
5	Localization Metastasis	Pulmonary	1

Table 5: Example data in EAV format.

This translation has been implemented as a function of the **TablePreprocessor** tool. The **TablePreprocessor** iterates over all Excel spreadsheet cells (as in Table 2), splits the value vectors and for each instance of a concept creates a new line in the EAV format (as illustrated in Table 3).

4.3 Lexical Matching (MDRMatcher)

To automatically identify correspondences between the source and target terminology, a lexical matcher (MDRMatcher) has been implemented.

It is based on a bag-of-words algorithm (Brownlee, 2017). A bag-of-words algorithm searches for all words from a source string in a target . The basic idea is to compare *all* concepts from the biobank (local) with *all* concepts of the central terminology (central, e.g. ADOPT CCDC) after all had been translated into a common language (such as English). The bag-of-words algorithm uses the designators of the concepts, i.e. the source and target strings. When a same word is found in *both* the target and source string, this contributes to a matching score, to estimate how "similar" two concepts are. The matches with the highest scores are then presented to the user as mapping suggestions. One extension to the bag-of-words algorithm we implemented is to also consider sequences of *n* subsequent words, also called *n*-grams (Jurafsky & Martin, 2009).

The MDRMatcher tool is also capable of simple synonym handling. For this, it has been extended with a publicly available "generic" list of medical abbreviations from Wikipedia[1] and specifically designed synonyms for the CCDC. These synonyms are stored in a customizable synonym file. The program is also capable of distinguishing between domain-specific medical terms (as

[1] https://en.wikipedia.org/wiki/List_of_medical_abbreviations

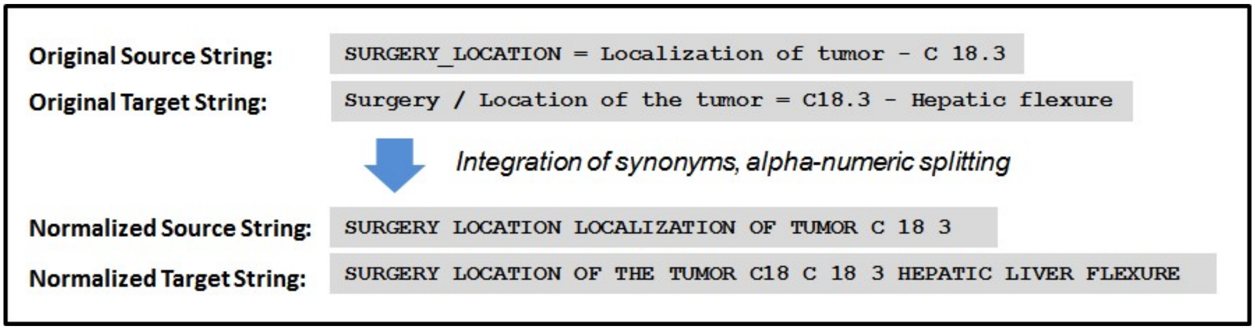
Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

used in the biobank data or in the CCDC terminology) and common/everyday English terms. For this, a database[2] of the most common 10.000 English words has been integrated into the MDRMatcher. The algorithm up-ranks medical terms and down-ranks common English terms.

The process works as follows:

As a preparatory step, the tool first normalizes all entries from the source and target terminologies, as illustrated in Figure 4. It does this by removing all non-alphanumeric characters and capitalizing the strings. It then processes synonyms by inserting them into the normalized strings. This is illustrated with "HEPATIC", which is replaced with "HEPATIC LIVER". The synonyms database also includes entries for “positive” and “negative” words, such as *False*, *No*, *Not*, *True* and *Yes*, which are augmented with the entries *neg* or *pos*. This theoretically allows for up-ranking "positive" to "positive" concepts, and "negative" to "negative" ones. It also splits strings whenever a numeric character directly follows an alphabetic character, or vice versa, as shown with "C18", which is transformed into "C18 C 18". As it will be shown below, this allows for matching "C 18.3" in the original source string with "C18.3", as a sequence of three words.



Original Source String: SURGERY_LOCATION = Localization of tumor - C 18.3

Original Target String: Surgery / Location of the tumor = C18.3 - Hepatic flexure

Integration of synonyms, alpha-numeric splitting

Normalized Source String: SURGERY LOCATION LOCALIZATION OF TUMOR C 18 3

Normalized Target String: SURGERY LOCATION OF THE TUMOR C18 C 18 3 HEPATIC LIVER FLEXURE

Figure 4: Normalization of the two strings "SURGERY_LOCATION = Localization of tumor - C 18.3" and "Surgery / Location of the tumor = C18.3 - Hepatic flexure".

The program then performs the actual matching. In contrast to typical a bag-of-words algorithm, which does not consider the word order, we also consider sequences of *n* subsequent words, also called *n*-grams (Jurafsky & Martin, 2009). The algorithm starts with the maximum number of possible words, ***WordCount***, which is defined as ***minimum(words(SourceString), words(TargetString))***. In the example from Figure 4, ***WordCount* = minimum(8, 12) = 8**. This value is then step-wise counted down to 1. In each iteration, all sequences of words (having this word count), are extracted and compared against the target string. Matches can either be an exact match (where both strings are equal) or a fuzzy match (both strings are similar). Whenever a match is found, it contributes to a final matching score, which has been implemented as follows:

[2] https://raw.githubusercontent.com/first20hours/google-10000-english/master/google-10000-english-usa.txt

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

- Exact match: ***CharacterLength * WordCount * 2***

- Fuzzy match:

  - If the first characters are equal:
    ((CharacterLength1 + CharacterLength2) / 2.0) * WordCount * Ratio

  - If the first characters are not equal:
    (((CharacterLength1 + CharacterLength2) / 2.0) * WordCount * Ratio) * 0.5

    with ***Ratio = 1 - LevenshteinDistance(SourceString, TargetString) / maximum(CharacterLength1, CharacterLength2)***

Note how exact matches are up-ranked (compared to fuzzy matches) by a multiplication with 2. A fuzzy match is reduced to half when the first characters of both strings are different. This prevents creating high scores when comparing entries such as "C18" and "D18". The constants and behavior of the algorithm were determined experimentally using the first data set obtained from the CCDC. Table 6 shows this step-wise decrease of the value ***WordCount*** (8...1), however only for values where an exact or fuzzy match was found. For example, in the first column, a sequence of 3 words, which is “C 18 3” was found both in the source and target string. It is an exact match and the score contribution is 36. In the same fashion, shorter sequences of words contribute to the overall matching score of the two terms shown in Figure 4, which is 191,49 (the sum of all score contributions).

WordCount	**In Source String**	**Exact (=) / Fuzzy (~)**	**In Target String**	**Score Contribution**
3	C 18 3	=	C 18 3	36.0
2	SURGERY LOCATION	=	SURGERY LOCATION	64.0
2	C 18	=	C 18	16.0
2	18 3	=	18 3	16.0
1	SURGERY	=	SURGERY	14.0
1	LOCATION	=	LOCATION	16.0
1	LOCALIZATION	~	LOCATION	6.66
1	OF	=	OF	4.0
1	TUMOR	=	TUMOR	10.0

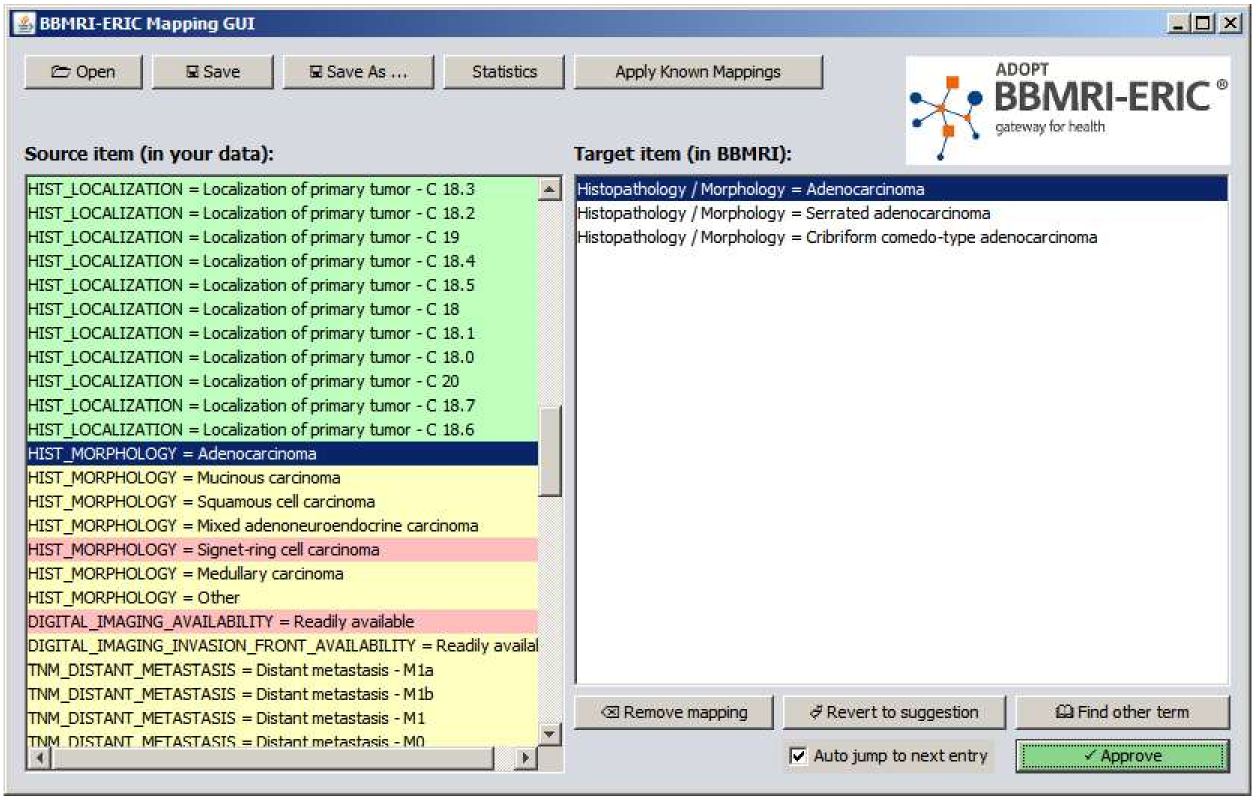
1	C	=	C	2.0
1	18	~	C18	0.83
1	18	=	18	4.0
1	3	=	3	2.0

Table 6: Computation of the matching score for the two normalized strings in Figure 4.

4.4 Curation of Mappings by Human Expert (Mapping GUI)

As it will be described in more detail in the results section, the MDRMatcher delivers good results and manages to align almost 80% of the terminology items fully automatically without user interaction. However, for the utilization in the ETL process, these mappings need to be validated and potentially corrected by a human expert.

Although the mapping file format (that is generated by MDRMatcher) has been designed to be easily editable with Excel, it was quickly observed that a dedicated user interface for the manipulation of the mapping files would speed-up the whole curating process. Based on these experiences, a graphical "Mapping GUI" program has been developed (see screenshot in Figure 5).



Source item (in your data):

```
HIST_LOCALIZATION = Localization of primary tumor - C 18.3
HIST_LOCALIZATION = Localization of primary tumor - C 18.2
HIST_LOCALIZATION = Localization of primary tumor - C 19
HIST_LOCALIZATION = Localization of primary tumor - C 18.4
HIST_LOCALIZATION = Localization of primary tumor - C 18.5
HIST_LOCALIZATION = Localization of primary tumor - C 18
HIST_LOCALIZATION = Localization of primary tumor - C 18.1
HIST_LOCALIZATION = Localization of primary tumor - C 18.0
HIST_LOCALIZATION = Localization of primary tumor - C 20
HIST_LOCALIZATION = Localization of primary tumor - C 18.7
HIST_LOCALIZATION = Localization of primary tumor - C 18.6
HIST_MORPHOLOGY = Adenocarcinoma
HIST_MORPHOLOGY = Mucinous carcinoma
HIST_MORPHOLOGY = Squamous cell carcinoma
HIST_MORPHOLOGY = Mixed adenoneuroendocrine carcinoma
HIST_MORPHOLOGY = Signet-ring cell carcinoma
HIST_MORPHOLOGY = Medullary carcinoma
HIST_MORPHOLOGY = Other
DIGITAL_IMAGING_AVAILABILITY = Readily available
DIGITAL_IMAGING_INVASION_FRONT_AVAILABILITY = Readily availal
TNM_DISTANT_METASTASIS = Distant metastasis - M1a
TNM_DISTANT_METASTASIS = Distant metastasis - M1b
TNM_DISTANT_METASTASIS = Distant metastasis - M1
TNM_DISTANT_METASTASIS = Distant metastasis - M0
```

Target item (in BBMRI):

```
Histopathology / Morphology = Adenocarcinoma
Histopathology / Morphology = Serrated adenocarcinoma
Histopathology / Morphology = Cribriform comedo-type adenocarcinoma
```

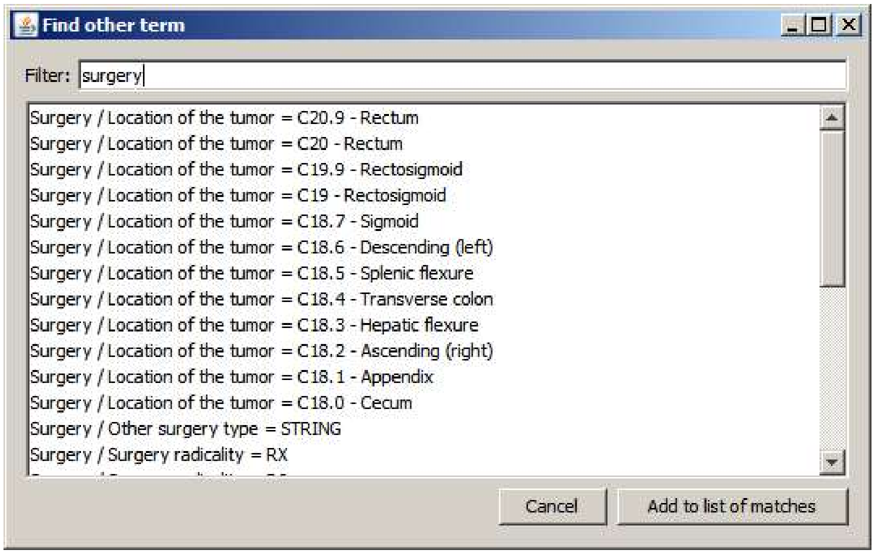
Figure 5: Mapping GUI program.

The program displays the source terminology from the biobank on the left, and the target CCDC terminology on the right part of the window. When a user selects a term on the left, the tool shows the corresponding CCDC mapping proposals from MDRMatcher on the right. These entries are sorted based on the matching score, which is hidden in the tool to avoid confusing the user.

The user can than either approve a mapping or correct it by selecting another term on the right side. It is also possible to remove a bad mapping or to revert to MDRMatcher's initial suggestion. If the correct term was not under the list of proposals, a search window can be opened, where the term can be looked up manually (see Figure 6). It is also possible to unselect any mapping via the "Remove mapping" button in the case when no mapping from the source terminology to the target terminology exists at all. In this case, the data of this source term is ignored during the actual data transformation in ETLHelper.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.



Filter: surgery

Surgery / Location of the tumor = C20.9 - Rectum
Surgery / Location of the tumor = C20 - Rectum
Surgery / Location of the tumor = C19.9 - Rectosigmoid
Surgery / Location of the tumor = C19 - Rectosigmoid
Surgery / Location of the tumor = C18.7 - Sigmoid
Surgery / Location of the tumor = C18.6 - Descending (left)
Surgery / Location of the tumor = C18.5 - Splenic flexure
Surgery / Location of the tumor = C18.4 - Transverse colon
Surgery / Location of the tumor = C18.3 - Hepatic flexure
Surgery / Location of the tumor = C18.2 - Ascending (right)
Surgery / Location of the tumor = C18.1 - Appendix
Surgery / Location of the tumor = C18.0 - Cecum
Surgery / Other surgery type = STRING
Surgery / Surgery radicality = RX

Figure 6: Mapping GUI program's target terminology search. Pressing the "Add to list of matches" button adds the selected term to the right side of the main window (Figure 5).

Another feature that was integrated into the program is an automatic backup and retrieval functionality of user-validated mappings. Whenever a biobank provides new data, typically a new mapping file has to be generated with MDRMatcher that contains the updated metadata. Unfortunately, this overwrites the already user-curated mappings; however, with the restore function, the known mappings can be restored easily via the "Apply Known Mappings" button. Then the user only has to curate new/changed items. This restore functionality also works across biobanks.

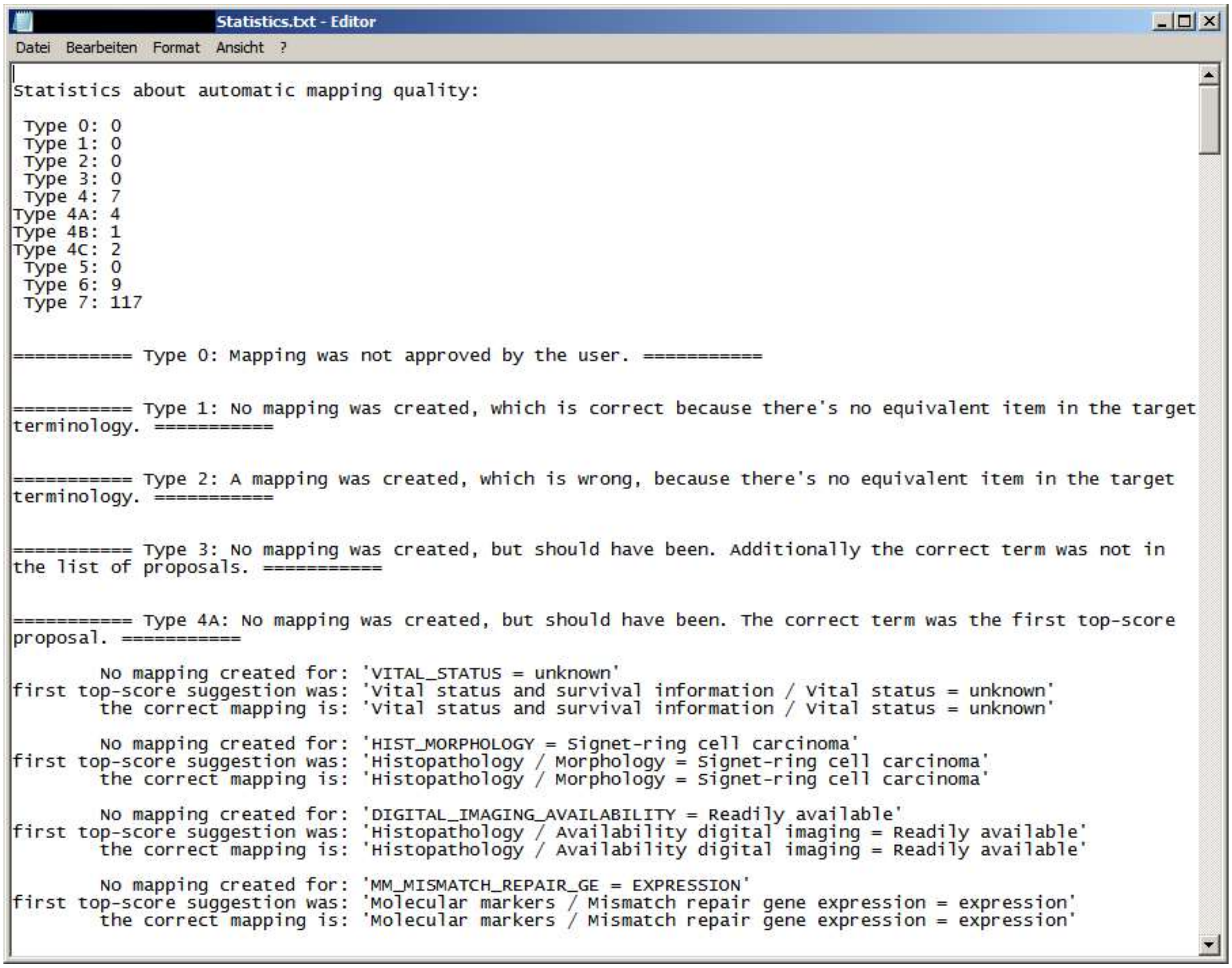
By copying all validated mappings from the "knownMappings" directories into the directory of the biobank currently being processed, these mappings can be applied to the current biobank via the "Apply Known Mappings" button. When another biobank used the same term for a data element and a mapping has already been validated, this mapping can then be applied to the biobank currently being processed. At the moment, this implementation can be considered experimental, especially since it also requires some manual work when copying the files. If it turns out that mappings can indeed be transferred for the most part between different biobanks, it might be worth considering storing them in a true central repository.

The tool also internally tracks the mapping status for each mapping, which is:

- 0: No mapping was or could be proposed
- 1: A mapping was proposed by MDRMatcher
- 2: The mapping was approved by a human expert

Terms in the source terminology are highlighted in different colors based on the mapping status (0 = red, 1 = yellow, 2 = green).

This information can also be used to compare the original mappings (as proposed by MDRMatcher) with the expert-curated mappings. This allows deriving statistics about the automatic mapping quality of MDRMatcher. By pressing the “Statistics” button, the program creates a text file (see Figure 7) where it lists all mapping, classified according the four-axial classification scheme that will be described in section 5.4, “Mapping Quality”. This file can be used to analyze common patterns of “misbehavior” in the matching algorithm. This file was used for the evaluation that is described in the results section.



```
Statistics about automatic mapping quality:

 Type 0: 0
 Type 1: 0
 Type 2: 0
 Type 3: 0
 Type 4: 7
Type 4A: 4
Type 4B: 1
Type 4C: 2
 Type 5: 0
 Type 6: 9
 Type 7: 117


=========== Type 0: Mapping was not approved by the user. ===========


=========== Type 1: No mapping was created, which is correct because there's no equivalent item in the target
terminology. ===========


=========== Type 2: A mapping was created, which is wrong, because there's no equivalent item in the target
terminology. ===========


=========== Type 3: No mapping was created, but should have been. Additionally the correct term was not in
the list of proposals. ===========


=========== Type 4A: No mapping was created, but should have been. The correct term was the first top-score
proposal. ===========

        No mapping created for: 'VITAL_STATUS = unknown'
first top-score suggestion was: 'Vital status and survival information / Vital status = unknown'
        the correct mapping is: 'Vital status and survival information / Vital status = unknown'

        No mapping created for: 'HIST_MORPHOLOGY = Signet-ring cell carcinoma'
first top-score suggestion was: 'Histopathology / Morphology = Signet-ring cell carcinoma'
        the correct mapping is: 'Histopathology / Morphology = Signet-ring cell carcinoma'

        No mapping created for: 'DIGITAL_IMAGING_AVAILABILITY = Readily available'
first top-score suggestion was: 'Histopathology / Availability digital imaging = Readily available'
        the correct mapping is: 'Histopathology / Availability digital imaging = Readily available'

        No mapping created for: 'MM_MISMATCH_REPAIR_GE = EXPRESSION'
first top-score suggestion was: 'Molecular markers / Mismatch repair gene expression = expression'
        the correct mapping is: 'Molecular markers / Mismatch repair gene expression = expression'
```

Figure 7: Statistics text file generated by the Mapping GUI program. For the types 0-7 see section 5.4, “Mapping Quality”.

4.5 Data Transformation (ETLHelper)

The last tool in the data harmonization pipeline is the **ETLHelper** tool, which takes the *Mapping File* (plus the *Local Metadata Definition File* and the *Central Metadata Definition File*) and the

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

CSV/EAV file as input and generates an XML file, which can be uploaded into the target system (currently the CCDC system, planned are the integration of the future BBMRI-ERIC Connector and the *German Biobank Alliance* system).

The **ETLHelper** tool processes the input data in combination with the mapping information. This processing comprises replacing the source value sets with those of the target terminology, as well as transforming data types ("castings"). The **ETLHelper** functionality is based on four steps (which are described in more detail in the following sections):

1. Upload of all files into a SQLite database (which is integrated into the **ETLHelper** environment)
2. Applying the mapping rules on the biobank data via SQL
3. Executing data type transformations ("castings") if a mapping connects two different data types (such as Boolean and Enumerated)
4. Generating the final XML file

4.5.1 Upload of Data into SQLite Database

It was decided to base large parts of the ETL on SQL technology. The reason for that is that SQL databases already implement routines to efficiently sort, filter, and align data in tabular format. While such functionality could have been implemented with custom code, relying on existing SQL technology appeared to be more appropriate and efficient.

The SQL database is only used internally (directly bundled within the **ETLHelper** environment) and remains completely hidden to the user. A biobank user does not have to interact with it.

For our purposes SQLite (https://www.sqlite.org/index.html) was chosen to be integrated with the **ETLHelper**. According to their website, "SQLite is a self-contained, high-reliability, embedded, full-featured, public-domain, SQL database engine. SQLite is the most used database engine in the world." The first feature is important, because this allowed an integration of the database into the **ETLHelper** tool.

As outlined in Figure 1 and in order to generate the XML file, the ETLHelper tool needs to process the following information:

- The CSV/EAV data from the biobank
- The mapping file
- The local metadata definition file (for the SOURCE SLOT field)

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

Figure 8 outlines the flow of information during the whole process. It does this by showing the most relevant table columns of the relevant input files in different colors, as well as their utilization by the different tools.

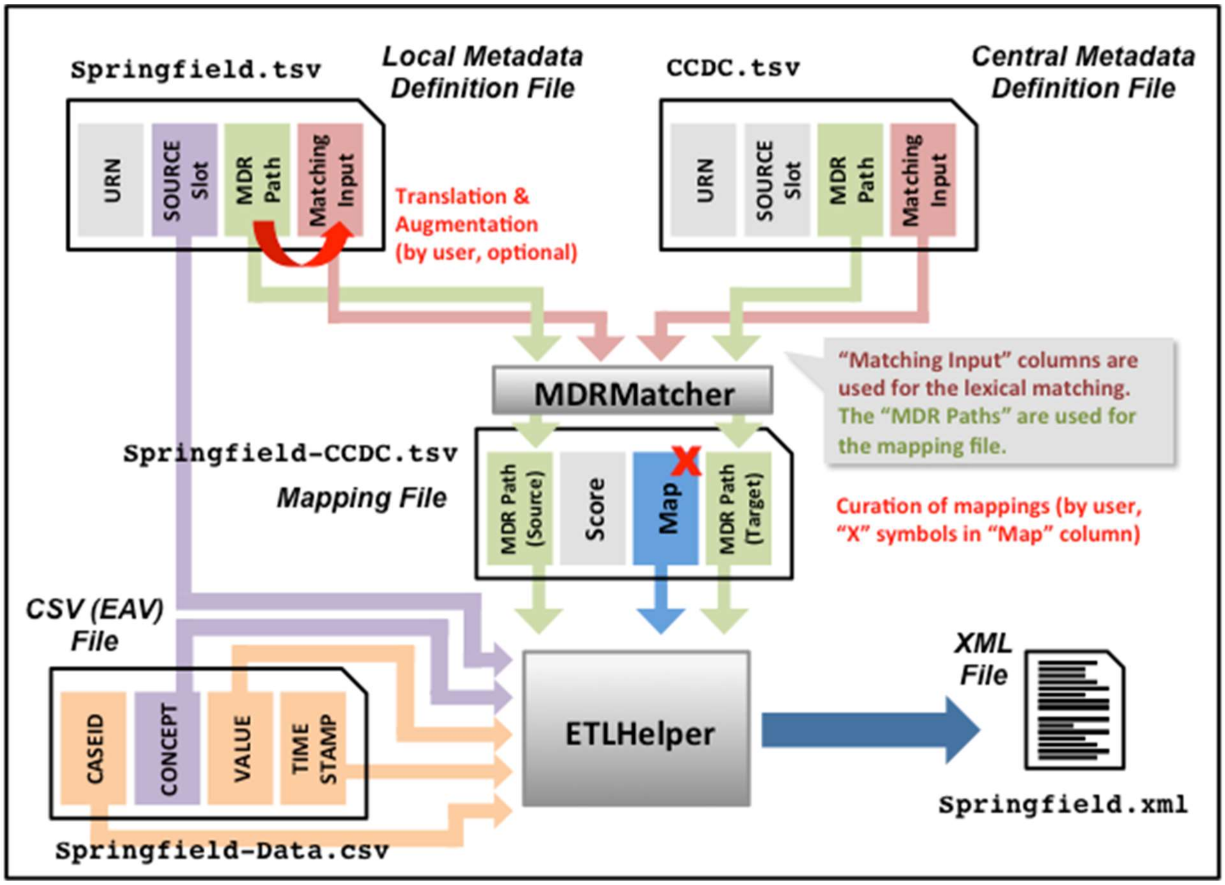

Springfield.tsv
Local Metadata Definition File
URN
SOURCE Slot
MDR Path
Matching Input
Translation & Augmentation (by user, optional)
CCDC.tsv
Central Metadata Definition File
URN
SOURCE Slot
MDR Path
Matching Input
MDRMatcher
"Matching Input" columns are used for the lexical matching. The "MDR Paths" are used for the mapping file.
Springfield-CCDC.tsv
Mapping File
MDR Path (Source)
Score
Map
MDR Path (Target)
Curation of mappings (by user, "X" symbols in "Map" column)
CSV (EAV) File
CASEID
CONCEPT
VALUE
TIME STAMP
Springfield-Data.csv
ETLHelper
XML File
Springfield.xml

Figure 8: Information flow between the tools/files.

To load the data into the program, an already existing implementation of a CSV file parser was integrated (https://viralpatel.net/blogs/java-load-csv-file-to-database/) and adapted to the ADOPT requirements.

4.5.2 Mapping of Biobank Data via SQL

When the information from the four input files are stored inside the SQLite database, the mapping rules, as found by MDRMatcher and as curated by the user, are applied to the real world input data. This works via a single SQL join statement that merges all necessary data columns from all input files, as outlined in Figure 8 with the arrows directed to the ETLHelper tool. For

Horizon 2020
ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

each actual patient data record in the input data, a corresponding record in the mapping rules with the same concept name, data type and data value is searched and - if found - applied. If no rule was found, the database table for the result of the mapping process contains null values for this particular input data record, indicating that no corresponding data element exists in the target namespace. Otherwise the values of the mapping rule (concept name, data type, and data value) are stored for this data record.

4.5.3 Data Type Transformations

The toolset has been designed around the semantics and architectural concepts of the Samply.MDR. These are in particular

- data types, and
- the hierarchical grouping of data elements.

Even though it is not necessary for a biobank to use the Samply.MDR when going the Excel file route, the Samply.MDR is used for the definition of the target terms. Therefore, the ETL tools have to adhere to these basic concepts of the Samply.MDR.

It is important to understand that the ETL tools internally handle metadata with what we call "metadata items". These are combinations of an attribute, its data type, and a value. These "metadata items" act as unambiguous and easily human-readable data element definitions.

As shown with examples in Table 7, for the Samply.MDR data types with an open data range (i.e., Integer, Float, String, Date, DateTime), we construct such metadata items by concatenating the attribute denominator with the data type while using an equal sign as separator, for example, "Glucose = INTEGER".

In contrast, for the Samply.MDR data types with a closed data range (i.e., Enumerated, Boolean), we directly specify the value instead of the data type, for example, "Vital Status = Alive" or "Patient is still alive = FALSE".

An attribute can be further extended with hierarchical information, e.g., "Patient Data / Demographics / Age = INTEGER", which puts the "Age" attribute into a class-like folder "Demographics", which is itself below the "Patient Data" folder. This is in particular important for the compatibility with the MDR. For example, the central CCDC metadata that is extracted from the MDR uses such paths. Note that these are not true sub-class definitions in an ontological correct way (i.e., they do not necessarily model "IS-A" relationships); they are a rather pragmatic way to organize/group attributes[3].

[3] As an example, consider the data element group „ Vital status and survival information " in the CCDC, which can be found here: https://mdr-test.ccp-it.dktk.dkfz.de/detail.xhtml?urn=urn%3Accdg%3Adataelementgroup%3A2%3A3. This group contains various data elements, such as "Timestamp of last update of vital status" or "Vital status". Although the members in this group are clearly related (that's why they are grouped in this way), the group cannot

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

MDR Data Type	**Example, as used in the source data**		**Metadata item (metadata description entry, as used in metadata / mapping files)**
	Attribute (EAV) or Column Caption (Excel)	**Value (EAV) or Cell Entry (Excel)**	
Enumerated	Vital Status	Alive	Vital Status = Alive
Integer	Glucose	100	Glucose = INTEGER
Float	PSA	5.5	PSA = FLOAT
Boolean (True)	Patient is still alive	TRUE	Patient is still alive = TRUE
Boolean (False)	Patient is still alive	FALSE	Patient is still alive = FALSE
String	Treatment	Radiation Therapy	Treatment = STRING
Date	Birthday	1983-03-19	Birthday = DATE
DateTime	Surgery Time	2015-05-01 17:03:00	Surgery Time = DATETIME

Table 7: Samply.MDR data types and examples for source data records and mappings.

These metadata items are used in metadata definitions files (as generated by TablePreprocessor, MDRExtractor and as used by MDRMatcher) and mapping files (as generated by MDRMatcher). Their structure makes them easily human-readable as they can be sorted alphabetically, which is useful when viewing larger collections. Most importantly, the lexical bag-of-words matcher benefits from their structure (going from the most general data element group, over the attribute down to the very specific data type or value) by partially considering the word sequence.

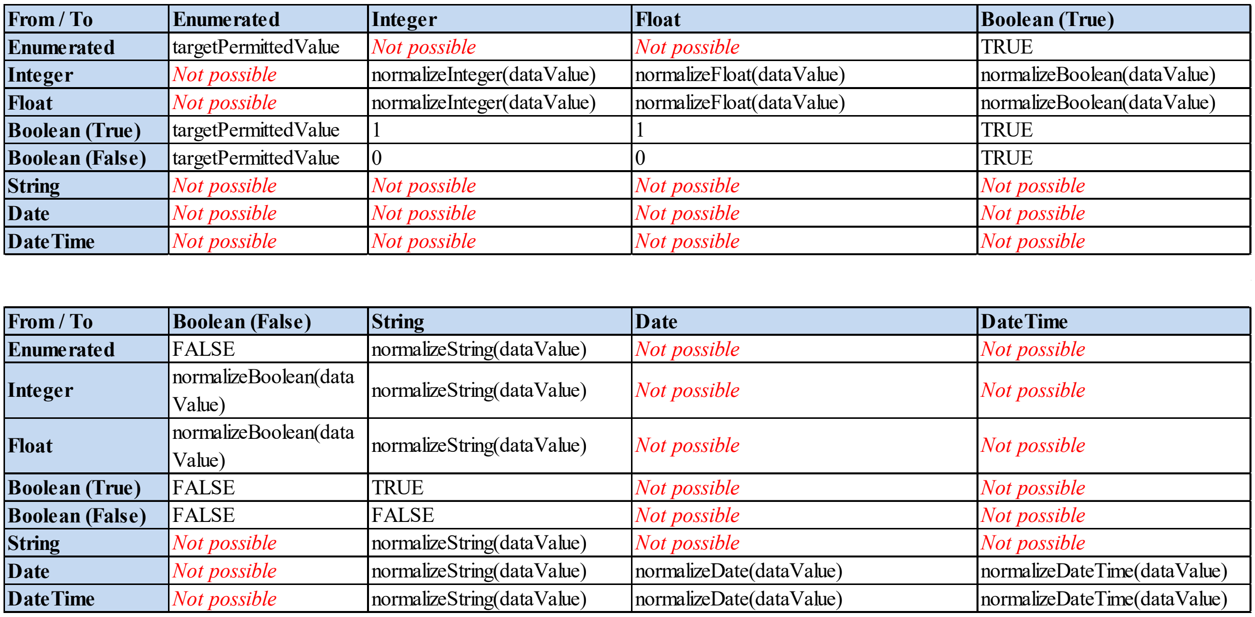
To continue with the example mapping from above (**Vital Status = Alive => Patient is still alive = TRUE**), this mapping asks to translate from an "Enumerated" datatype (with the value "Alive") into a Boolean data type (with the value "TRUE"). In other words, this means that whenever the ETL tool encounters a "Vital Status" concept with the value "Alive", it is supposed to generate a "Patient is still alive" Boolean concept with the value TRUE.

This process involves the translation, or casting, between data types (here between “Enumerated” and “Boolean”). Our implementations of the translation rules from one data type into another are shown in Figure 9. The leftmost column indicates the data type to be transformed, the top row the target data type into which it is to be transformed. The cell values tell whether a transformation is possible or not. In the latter case, this is indicated by the "Not possible" entries.

be considered a true “class” in an ontological way, nor can one say that the “Timestamp of last update of vital status” data element is a specialization of “Vital status and survival information”.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.



From / To	Enumerated	Integer	Float	Boolean (True)
Enumerated	targetPermittedValue	Not possible	Not possible	TRUE
Integer	Not possible	normalizeInteger(dataValue)	normalizeFloat(dataValue)	normalizeBoolean(dataValue)
Float	Not possible	normalizeInteger(dataValue)	normalizeFloat(dataValue)	normalizeBoolean(dataValue)
Boolean (True)	targetPermittedValue	1	1	TRUE
Boolean (False)	targetPermittedValue	0	0	TRUE
String	Not possible	Not possible	Not possible	Not possible
Date	Not possible	Not possible	Not possible	Not possible
DateTime	Not possible	Not possible	Not possible	Not possible

From / To	Boolean (False)	String	Date	DateTime
Enumerated	FALSE	normalizeString(dataValue)	Not possible	Not possible
Integer	normalizeBoolean(dataValue)	normalizeString(dataValue)	Not possible	Not possible
Float	normalizeBoolean(dataValue)	normalizeString(dataValue)	Not possible	Not possible
Boolean (True)	FALSE	TRUE	Not possible	Not possible
Boolean (False)	FALSE	FALSE	Not possible	Not possible
String	Not possible	normalizeString(dataValue)	Not possible	Not possible
Date	Not possible	normalizeString(dataValue)	normalizeDate(dataValue)	normalizeDateTime(dataValue)
DateTime	Not possible	normalizeString(dataValue)	normalizeDate(dataValue)	normalizeDateTime(dataValue)

Figure 9: Transformation rules between different data types.

For all cases, where a data transformation is possible, there are two basic kinds. The first one involves generating a new target value (while ignoring the source value). For example, when transforming from “Enumerated” to “Enumerated”, the new data record should use the permitted value of the target term (called “targetPermittedValue” in Figure 9) - the source value is basically removed. As explained above, this happens on the value level. As an example, consider the following two mappings:

Source Metadata Item	Target Metadata Item
SURVIVAL_STATUS = LIVING	Vital Status = Person Is Still Alive
SURVIVAL_STATUS = DECEASED	Vital Status = Person Is Dead

When performing the ETL, the rule is to replace all “LIVING” values (as they appear in a source data file from a biobank) with the *targetPermittedValue*, which is “Person Is Still Alive” for the first mapping above (“living”).

Similarly, when translating a “Boolean (True)” into an “Integer”, it uses the new value “1” as shown in Figure 9. In contrast, the other kind uses the original source value to derive a new output value. This requires an additional check whether the source value is a valid instance of its data type. As shown in the table, for example, when translating from an Float to an Integer, a function normalizeFloat() is called to check whether the data record is a floating point value. It can then be casted by the Java program code into an integer afterwards.

4.5.4 Generating the Final XML File

The final processing step in ETLHelper is the creation of the XML file. After the above steps, the fully integrated data is available in ETLHelper's SQLite database. The creation of the XML files works by instantiating Java beans, for which Java classes were generated out of the XSD schema provided by WP2. The Java beans are then serialized into the final XML file.

The ETL process described in chapter 4.5 deals with the semantic aspects of the data harmonization. Regarding syntactic/structural aspects, the tool currently only supports the creation of XML files for the CCDC system. However, it is possible to integrate the creation of other XML formats, as long as a XSD specification is available. This step is not generic and requires implementing Java code. We already integrated an experimental version that generates XML files according to a specification from the German Biobank Alliance project. In later stages of BBMRI-ERIC, the tool needs to be extended to support the BBMRI-ERIC Connector.

4.6 Conversion of the XML File into the Legacy Format

The ELHelper tool described above generates XML files that comply with a new format that has been developed during a hackathon between CS-IT WP2 and ADOPT WP3 / CS-IT WP8 in Erlangen in December 2017. Unfortunately, support for this new format was not integrated into the (then already running) CCDC system. As a work-around, an XML converter has been developed (by author PH), which was subsequently extended to meet the requirements of the ETL tools and the target terminology.

5 Results

This section outlines the preliminary results from the CCDC, for which the toolset was used. At the time of the writing of this document, the data from ten biobanks has been processed.

5.2 Workflow of ETL for the CCDC

The overall ETL workflow was as follows:

1. The biobank extracted the data out of its source system and converted it into the Excel format. This Excel file was uploaded by the biobank via the CCDC uploader, which has been developed by BBMRI-ERIC CS IT WP2.
2. BBMRI-ERIC CS IT WP8 was notified whenever new uploads took place. It downloaded the file and performed some basic checks to determine the overall suitability of the data for the CCDC.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

3. The biobank was contacted by BBMRI-ERIC CS IT WP8 and problems with the data were discussed. In the case of minor (easily correctable) issues, BBMRI-ERIC CS IT WP8 solved these; in the case where major changes were necessary, the biobank was asked to provide a new file via the above-mentioned CCDC uploader. Such errors include overall formatting errors.
4. After that the Excel data was fed into the ETL pipeline by BBMRI-ERIC CS IT WP8, that is:
    - assigning the data types to the Excel spreadsheet columns,
    - converting the Excel file into the EAV format (TablePreprocessor),
    - performing the lexical matching against the CCDC terminology (MDRMatcher),
    - with MappingGUI, importing already curated mappings from other biobanks (as far as this is possible) and curating the remaining mappings,
    - executing the actual data transformation (ETLHelper), and
    - legacy XML format conversion.
5. Often the use of MDRMatcher and ETLHelper identified additional issues with the data, such as additional formatting errors or shifted cell entries in the Excel file. In this case, the biobank was contacted to correct these. In many cases the problems could be corrected by BBMRI-ERIC CS IT WP8, in some a new upload was necessary. These were often not problems with data quality, but questions regarding content, i.e. how a data element was to be understood.
6. Interim results were constantly discussed with the biobank. This includes the previously created mapping files as well as the ETL reports generated later.
7. The XML files generated by BBMRI-ERIC CS IT WP8 were collected and uploaded again (in a bundled ZIP file) via the CCDC file uploader. BBMRI-ERIC CS IT WP2 then checked the compatibility and uploaded the data into the CCDC, where it also performed completeness checks.

5.3 Common Patterns of Data Received

During the execution of the CCDC ETL, different types of problems with the biobank data were observed. Some of them were detected directly by BBMRI-ERIC CS IT WP8 upon receiving the Excel file; some were later detected by the ETL tools. In particular, the TablePreprocessor tools allows observing shifted cell entries and otherwise incorrect value sets. The ETLHelper tool on the other hand outputs comprehensive statistics about the data and potential errors that may happen during the ETL. The Listings 1 and 2 in the Appendix contain sample log files from these two programs.

As a summary, common problems with the Excel data received included:

- Shifted data in the Excel file (possibly copy & paste errors)
- Empty columns (no values) in Excel file

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

- Non-uniform upper and lower case, spaces at the beginning and end of the entries
- "Unknown", "not available" or "N/A" as entries, which did not contribute data to the CCDC and therefore had to be rejected.
- Additional (free-text) comments, e.g. for data elements where an integer number was requested.
- Comma instead of semicolon to separate instances
- Non-standard characters, e.g. the non-standard dash character (–) instead of the standard dash (-), which was possibly caused by Microsoft Office's auto formatting. This also includes apparently/visually normal characters that were actually non-standard characters.
- Utilization of unclear abbreviations, e.g. "ND" = "not done", ".A" or ".U", "SX", "DX"
- Confusion with date-related entries, e.g. provision of a date when a relative date was requested.
- Redundant data elements, i.e., duplicate columns
- Provision of non-speaking value sets (e.g. "1" and "2") without mentioning its meaning (e.g. "1"="male", "2"="female").
- Inconsistent use of language (e.g., partially English, partially other language)

5.4 Mapping Quality

During the execution of the ETL, the MDRMatcher program has been used to automatically generate mapping proposals. To assess the quality of the (automatically proposed) mappings from MDRMatcher, the Mapping GUI program was extended in a way that it internally keeps the original mapping (the one proposed by MDRMatcher). After the user has done the corrections to the mappings (where necessary), this allows for comparing the “old” automatic mappings with the user-curated mappings.

To classify the behavior of MDRMatcher on a per-concept basis, a four-axial classification scheme has been developed. It is comprised out of the four dimensions *conceptual*, *mapping*, *correctness* and *matching:*

Conceptual dimension:

- Source data element **has an** equivalent in the target terminology
- Source data element **has no** equivalent in the target terminology

Mapping dimension:

- MDRMatcher has created a mapping
- MDRMatcher has not created a mapping (because there are several top matches with the same score)

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

Correctness dimension:

- The behavior of MDRMatcher was **correct**
- The behavior of MDRMatcher was **wrong**

Matching dimension:

- The correct item **is** among the matches (proposed mappings)
- The correct item **is not** among the matches

The *conceptual dimension* describes whether the source data element has an equivalent in the target terminology. This fact can by analyzed by examining if the user (while curating the automatically generated mappings) has kept an existing mapping, created a new one or has removed the mapping. This dimension is completely independent of the behavior of the software; rather, it describes on a conceptual level whether a mapping for a data element from the source terminology can be created at all. If no mapping should have been created (conceptual), but the software created one, this has further implications. These are then described by the other axes. But if there is such an equivalent data element the matcher should have created a mapping. This information is captured in the *mapping dimension*. Note that the MDRMatcher always finds a mapping, i.e. the "best" mapping. It does this by considering the top-score match as mapping, unless multiple matches share the same top score. The *correctness dimension* simply evaluates whether this top mapping is the correct one, or not. This is done by comparing the MDRMatcher-proposed mapping with the user-curated mapping. Finally, the *matching dimension* determines whether the correct term from the target terminology was among the list of shown matches, or whether it has been cut-off because its score was below the threshold.

Note that the combination of the four dimensions theoretically results in 2^4 = 16 classes. However, some combinations do not make sense, e.g., a created mapping cannot be correct if there's no equivalent term in the target terminology. Only seven classes are valid for the classification, as shown in Table C. In addition, a class “0” was introduced, which counts all source items for which no mapping has been approved by the user. These include items, for which it was not clear where they should be mapped to, or they were deliberately excluded from the mapping. While this does not directly indicate the data quality of the biobank’s data, it provides a rough estimation on the effort that was put into compiling the data.

The applicable classes are:

- 0: Mapping has not been approved by a human expert.
- 1: No mapping was created, which is correct because there's no equivalent item in the target terminology.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

- 2: A mapping was created, which is wrong, because there's no equivalent item in the target terminology.
- 3: No mapping was created, but should have been. Additionally the correct term was not in the list of proposals.
- 4: No mapping was created, but should have been. The correct term was in the list of proposals.
    - 4A: It was the first top-score proposal.
    - 4B: It was one of the other top-score proposals.
    - 4C: It was one of the lower-score proposals.
- 5: A wrong mapping was created. The correct one was not in the list of proposals.
- 6: A wrong mapping was created. The correct one was in the list of proposals.
- 7: A correct mapping was created.

The classes 4A, 4B and 4C further specify class 4 by distinguishing where the correct term was in the list of proposals. Note that "top-score" proposals are all matches that share the same score value.

The results for the ten biobanks (as of August 2018) are summarized in Table 8. As it can be observed, 78.75% of all automatically generated mappings (by MDRMatcher) were correct (types 1 and 7). Only for less than 2% (types 2, 3, and 5), the correct mapping was not in the list of proposed mappings and had to be searched manually. For the remaining 19.66% (types 4 and 6), the correct mapping was under the list of suggestions and could be selected easily.

As a next step, it should be examined why certain errors happened, especially:

- Why a wrong target term has been selected (type 6)
- Why a correct term was not in the list of suggestions (types 3 and 5)

Note that the explanation for cases where no mapping was created at all (types 1, 3 and 4) is simple. The MDRMatcher does not create a mapping whenever more than one of the possible matches share the same top matching score (e.g., the top three matches share the same top score). In this case, the program is simply unable to determine which one of the matches is the "better" one and leaves this decision to the user. However, it is worth examining why a term did not appear in the list of suggestions (types 3, 5 and 6). We plan to further examine this being part of a future scientific publication. The findings will enable us to further refine the matching algorithm in MDRMatcher in the future.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

Type	Equivalent in Target	Created A Mapping	Correct Behaviour	Among Proposals	Explanation / Details	Biobank #1	Biobank #2	Biobank #3	Biobank #4	Biobank #5	Biobank #6	Biobank #7	Biobank #8	Biobank #9	Biobank #10	% of total
0	N/A	N/A	N/A	N/A	Mapping has not been approved by a human expert.	0	0	3	0	0	0	0	0	0	0	0,21
1	No	No	Yes	No	No mapping was created, which is correct because there's no equivalent item in the target terminology.	10	1	7	0	0	0	0	0	0	0	1,27
2	No	Yes	No	No	A mapping was created, which is wrong, because there's no equivalent item in the target terminology.	2	2	1	0	1	0	0	0	1	0	0,50
3	Yes	No	No	No	No mapping was created, but should have been. Additionally the correct term was not in the list of proposals.	0	0	0	0	0	0	0	1	0	0	0,07
4	Yes	No	No	Yes	No mapping was created, but should have been. The correct term was in the list of proposals.	17	12	36	12	14	7	17	18	14	8	10,98
4A	Yes	No	No	Yes	It was the first top-score proposal.	8	1	9	2	3	4	4	4	2	4	2,90
4B	Yes	No	No	Yes	It was one of the other top-score proposals.	5	5	7	3	11	1	5	4	4	0	3,19
4C	Yes	No	No	Yes	It was one of the lower-score proposals.	4	6	20	7	0	2	8	10	8	4	4,89
5	Yes	Yes	No	No	A wrong mapping was created. The correct one was not in the list of proposals.	2	1	2	2	0	0	1	2	1	1	0,85
6	Yes	Yes	No	Yes	A wrong mapping was created. The correct one was in the list of proposals.	11	10	14	8	12	9	18	12	18	10	8,64
7	Yes	Yes	Yes	Yes	A correct mapping was created.	163	83	82	94	90	117	142	117	96	110	77,48
					Total:	205	109	145	116	117	133	178	150	130	129	100,00

Table 8: Mapping Results. The numbers are referring to distinct value-level mappings. Green indicate the percentage of mappings for which the automated mapping was fully correct.

5.5 ETL Results

The mappings, proposed by MDRMatcher and user-curated within the MappingGUI program, were then used to convert the actual biobank data from the Excel format into the XML import format. Table 9 below is based on the statistics generated by ETLHelper and summarizes the ETL process for the data of the ten CCDC biobanks. The table is partitioned into three sections, which correspond to the three basic ETL steps that were described in sections 4.5.1 to 4.5.3:

- **Input data:** These statistics derive from what was described above in section “4.5.1 Upload of Data into SQLite Database”. During the upload of the data, the ETLHelper counts the number of patients and the **number of data records**, which is the number of non-empty cells in the Excel file. The **number of different concepts** is the number of columns in the Excel file.
- **Merging of data:** After the upload of the data, and as described in section “4.5.2 Performing mapping of biobank data via SQL”, the program evaluates for how many of the data records a mapping from the mapping file can be used to transform the actual data entries.
- **Data transformation:** For those data records, for which a mapping rule is available, data castings have to be applied to transform the data record into the target representation, as described in section “4.5.3 Executing data type transformations”. Table ETL1 summarizes the number and types of data castings and whether an error happened during the transformation, errors are in red.

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

Property			#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Input data	Biobank Data (Input Data)	Number of patients	1066	137	50	55	300	600	300	308	239	260
		Number of data records	50020	5112	2067	2477	11984	22617	13225	13729	9803	8840
		Number of different concepts	53	39	49	47	47	43	49	52	39	35
	Mapping Rules	Number of different source strings (dataelement and value)	205	109	145	116	117	133	178	150	130	120
		Number of found mappings between source and target strings	193	106	134	116	116	133	178	150	129	120
Merging of data	Mapping Complete-ness	Number of data records that should have a mapping	50020	5112	2067	2477	11984	22617	11325	13729	9803	8840
		Number of data records that do have a mapping	49263	4701	1932	2477	11983	22617	11325	13729	9774	8840
		Number of data records that do have a mapping (in %)	98,49%	91,96%	93,47%	100,00%	99,99%	100,00%	100,00%	100,00%	99,70%	100,00%
		Number of data records that don't have a mapping	757	411	135	0	1	0	0	0	29	0
Data transformation	Data Type Castings (OK/ERR)	ENUMERATED => ENUMERATED	35786/0	4012/0	1284/0	1907/0	9436/0	19350/0	10294/0	10798/0	8340/0	6499/0
		ENUMERATED => BOOLEAN	1119/0		100/0	28/0				308/0	239/0	263/0
		INTEGER => INTEGER	7093/**479**	274/0	230/0	300/**77**	995/0	2568/**1**	2140/0	1627/0	478/0	1038/0
		FLOAT => INTEGER		51/0			300/0					
		STRING => STRING	2654/0	137/0	126/0	55/0	636/0	98/0	191/0	380/0	239/0	520/0
		DATE => DATE	2132/0	227/0	92/0	110/0	652/0	600/0	600/0	616/0	457/**21**	518/**2**
	Summary of Castings	Number of transformation rules	49263	4701	1832	2477	11983	22617	13225	13729	9774	8840
		Number of errors	479	0	0	77	0	1	0	0	21	2
		Number of good transformations	48784	4701	1832	2400	11983	22616	13225	13729	9753	8838
		Number of good transformations (in %)	99,03%	100,00%	100,00%	96,89%	100,00%	100,00%	100,00%	100,00%	99,79%	99,98%
ETL Total Summary		Number of input data records that could be mapped and transformed (in %)	97,53%	91,96%	88,63%	96,89%	99,99%	100,00%	100,00%	100,00%	99,49%	99,98%

Table 9: ETL results, as reported by ETLHelper.

The last row, “ETL Total Summary”, outlines the total percentage of data records (that is, non-empty Excel cells) that could be transformed into the target (CCDC) representation. The ETL process delivered overall good results, with most biobanks achieving a transformation rate of or close to 100%.

However, evaluating what went wrong in the other cases will enable improving the ETL in the future. The ETLHelper tool therefore was extended to generate additional statistics to enable the closer examination of such cases.

The following outlines a few examples:

Biobank #1: Contains a data element “Targeted Therapy Scheme” with nine enumerated values, which could not be mapped as there are no corresponding data elements in the CCDC terminology. The 479 errors in the INTEGER => INTEGER type casting were triggered by entries, which are not valid integer values (e.g. “unknown”).

Biobank #2: Three mappings could not be created because three columns in the Excel file only contained “Unknown” entries. The target terminology, however, only allows a predefined value list, which does not include “Unknown”. During the mapping process it was therefore decided not to map these entries.

Biobank #3: Similar to biobank #2, this one’s data contained eight data elements with “ND” and “N/A” entries.

Biobank #4: The 77 errors in the INTEGER => INTEGER type casting were triggered by “No targeted therapy” entries, which are of course not valid integer values.

Biobank #9: Similar to biobank #4, this biobank used strings for date fields with the entry “UNKNOWN”, which caused the DATE => DATE type casts to throw errors.

5 Discussion & Outlook

This deliverable describes the tools which have been developed as part of the BBMRI_ERIC ADOPT Ontology-Based Toolset for Mapping of the Biobanking Terminologies, as well as the process and workflow how to apply those tools in a concrete ETL process.

As we have shown within the exemplary ETL processes the tools delivered overall good results. On the one hand, the lexical matching process in MDRMatcher does much of the work of identifying correct mappings. After all, almost 80% are automatically mapped correctly - the user in-

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

terface of the "Mapping GUI" program supports the user with the remaining 20%. On the other hand, we could show that our generic approach of data type conversion in ETLHelper is suitable for performing the subsequent data transformations, but also for detecting errors in the source data.

The procedure presented here is generic and can theoretically be used in the context of the Connector without making any modifications. However, using the Connector will have different requirements, and in general this does not mean that the overall approach cannot be improved. The CCDC data processed so far were extracted quite selectively from the source systems by the biobanks and thus already largely corresponded to the CCDC data set. Using the toolset for the connector will potentially lead to new challenges because of the expected higher heterogeneity of the source data. However, it is difficult to give an accurate prognosis because neither the contents and the information model of the Connector are known yet, nor an overview of the data in the biobanks (source systems, format, contents, data quality, etc.) exists.

Nevertheless, it makes sense to consider in advance how the procedure can be improved. Here are some starting points:

The approach described here is basically based on the Samply MDR, in which the metadata of the source and target systems are described "formally", which then facilitates an automatic ETL approach. The workaround we implemented to extract the metadata directly from the Excel file might no longer be sufficient in the connector context. Due to the increased complexity and heterogeneity of connector data, semantically richer metadata may be required. The use of such metadata is already possible, but was not applied in the CCDC as such additional metadata could not be provided via the MDR by the biobanks.

Second, it needs to be discussed how larger metadata collections are being handled. During the CCDC it was found that some biobanks have several different data collections that somehow have to be merged in the connector. Here one would have to consider whether to store the metadata in separate namespaces or proceed differently.

The matching algorithm has not been adjusted throughout the CCDC to provide consistent results. However, during the CCDC, certain peculiarities in the behavior of the matcher became apparent, which should be taken into account in future versions. For example, the presence of validated mappings from the CCDC offers great potential for further improvement of the algorithm. One could automatically check whether changes to the algorithm (e.g. to the constants, see Chapter 4.3) lead to a larger number of correct mapping proposals. Or one could even extend the procedure so far that it “learns” iteratively by itself: Whenever mappings are cured, the mapping algorithm tries to adapt to the best constants/weights. This would then potentially improve the remaining mapping proposals.

Such an approach would have enormous potential if it worked across biobanks. However, this would require that the source data be similar and comparable, which cannot yet be estimated.

Although it was decided for the CCDC that the ETL approach should be locally executable, aspects of centralization become important again in this context. For example, it would be necessary to clarify not only how the parameters for the matching algorithm are synchronized between sites, but also how and where the approved mappings are stored. This would also allow a more efficient reusability of curated mappings than is now possible (as described in the text, this is even possible now, but files have to be exchanged manually).

Consideration must also be given to the data transformation part. Although, as mentioned above, the procedure is generic, it makes sense to investigate whether and how a support for standardized data elements (e.g. ICD codes) can be implemented. Here one could use already existing ontologies (e.g. UMLS, OxO (Jupp et al., 2017)) for mapping and then would not have to map these data elements to each other with the lexical matcher (which, however, would be theoretically feasible).

The method presented here allows only 1:1, but no n:1 mappings, i.e. there is no possibility to combine several data elements in the source data set to one data element in the target data set. This was sufficient for the CCDC, since smaller calculations (e.g. age at diagnosis, which is calculated from the birthday and the diagnosis date) could be done directly in the Excel file. This topic has already been investigated in the past (see (Mate et al., 2015)); GBA also discusses the need to extend the MDR to include transformation rules. But also here it applies that one must know the requirements and the source data first.

Finally, since the implementation of the ETL tools (with the exception of the MDMatcher) has not yet been completed during the implementation of the CCDC, the tools were not yet rolled out to actual biobankers. The ETL was carried out exclusively by the developers of the ETL software. As far as user-friendliness is concerned, practical experience is still lacking at the moment. This is to change in the future, and we are now working on evaluating the tools when being applied by non-computer scientists.

6 References

Brownlee, J. (2017, October). A Gentle Introduction to the Bag-of-Words Model. Retrieved September 5, 2018, from https://machinelearningmastery.com/gentle-introduction-bag-words-model/

Jupp, S., Liener, T., Sarntivijai, S., Vrousgou, O., Burdett, T., & Parkinson, H. E. (2017). OxO - A Gravy of Ontology Mapping Extracts. *Icbo*.

Jurafsky, D., & Martin, J. H. (2009). Speech and Language Processing. Prentice Hall.

Kadioglu, D., Weingardt, P., Ückert, F., & Wagner, T. (2016). Samply.MDR – Ein Open-Source-Metadaten-Repository. Presented at the SamplyMDR - Ein Open-Source-Metadaten-Repository, German Medical Science GMS Publishing House. http://doi.org/10.3205/16gmds149

Lablans, M., Kadioglu, D., Muscholl, M., & Ückert, F. (2015). Exploiting Distributed, Heterogeneous and Sensitive Data Stocks While Maintaining the Owner's Data Sovereignty. *Meth-*

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

ods Inf Med, *54*(4). http://doi.org/10.3414/ME14-01-0137

Lubke, S. (2017, August 31). *Matching as a Service - Maschinelle Übersetzung zum Matching medizinischer Terminologien*.

Mate, S., Kadioglu, D., Majeed, R. W., Stöhr, M. R., Folz, M., Vormstein, P., et al. (2017a). Proof-of-Concept Integration of Heterogeneous Biobank IT Infrastructures into a Hybrid Biobanking Network. *Stud Health Technol Inform*, *243*, 100–104.

Mate, S., Köpcke, F., Toddenroth, D., Martin, M., Prokosch, H.-U., Bürkle, T., & Ganslandt, T. (2015). Ontology-Based Data Integration Between Clinical and Research Systems. *PloS One*, *10*(1), e0116656. http://doi.org/10.1371/journal.pone.0116656

Mate, S., Vormstein, P., Kadioglu, D., Majeed, R. W., Lablans, M., Prokosch, H.-U., & Storf, H. (2017b). On-The-Fly Query Translation Between i2b2 and Samply in the German Biobank Node (GBN) Prototypes. *Stud Health Technol Inform*, *243*, 42–46.

Nadkarni, P. M., Marenco, L., Chen, R., Skoufos, E., Shepherd, G. M., & Miller, P. (1999). Organization of Heterogeneous Scientific Data Using the EAV/CR Representation. *J Am Med Inform Assoc*, *6*(6), 478–493.

Proynova, R., Alexandre, D., Lablans, M., van Enckevort, D., Mate, S., Eklund, N., et al. (2017). A Decentralized IT Architecture for Locating and Negotiating Access to Biobank Samples. *Stud Health Technol Inform*, *243*, 75–79. http://doi.org/10.3233/978-1-61499-808-2-75

Steffens, M., Husmann, G., Koca, M., Lablans, M., Komor, M., Zeissig, S., et al. (2012). IT behind a Platform for Translational Cancer Research - Concept and Objectives. *Stud Health Technol Inform*, *180*, 1135–1137. http://doi.org/10.3233/978-1-61499-101-4-1135

Storf, H., Schaaf, J., Kadioglu, D., Göbel, J., Wagner, T. O. F., & Ückert, F. (2017). [Registries for rare diseases : OSSE - An open-source framework for technical implementation]. *Bundesgesundheitsblatt - Gesundheitsforschung - Gesundheitsschutz*, *60*(5), 523–531. http://doi.org/10.1007/s00103-017-2536-7

van Ommen, G.-J. B., Törnwall, O., Bréchot, C., Dagher, G., Galli, J., Hveem, K., et al. (2015). BBMRI-ERIC as a Resource for Pharmaceutical and Life Science Industries: The Development of Biobank-Based Expert Centres. *European Journal of Human Genetics*, *23*(7), 893–900. http://doi.org/10.1038/ejhg.2014.235

7 Appendix

Listing 1: Report file from the TabelPreprocessor program.

```
Welcome to the Table Preprocessor Utility 1.0!

Found 55 columns in file:

 - Treating the first column (PAT_Number) as column with patient ids.
 - DIAG_CT_DONE = ENUMERATED
   - CT- Not done
   - CT - Done, data available
   - CT - Done, data not available
   - CT - Unknown
```

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

```
 - DIAG_MRI_DONE = ENUMERATED
   - MRI - Done, data not available
   - MRI - Done, data available
   - MRI - Not done
   - MRI - Unknown
 - DIAG_X_DONE = ENUMERATED
   - Lung imaging  - Done, data available
   - Lung imaging - Not done
   - Lung imaging - Unknown
   - Lung imaging - Done, data not available
 - DIAG_ Liver imaging _DONE = ENUMERATED
   - Liver imaging - Unknown
   - Liver imaging - Unknown, Not done
   - Liver imaging - Done, data available
   - Liver imaging - Done, data not available
 - DIAG_COLONOSCOPY = ENUMERATED
   - Colonoscopy diagnostic exam - Not done
   - Colonoscopy diagnostic exam - Positive
   - Colonoscopy diagnostic exam - Negative
• • •

- TARGETED_THERAPY_SCHEME = ENUMERATED
   - 2
   - Vectibix
   - IGN-CN
   - Rituximab
   - Panorex
   - Tivozanib
   - Avastin
   - Erbitux
   - unknown
 - TARGETED_THERAPY_END_RELATIVE = INTEGER
 - TARGETED_THERAPY_START_RELATIVE = INTEGER
 - OVERALL_SURVIVAL_STATUS = INTEGER
 - VITAL_STATUS_TIMESTAMP = DATE
 - VITAL_STATUS = ENUMERATED
   - DEATH_UNKOWN_REASON
   - DEATH_UNKNOWN_REASON
   - ALIVE
   - DEATH_COLON_CANCER
   - DEATH_Unknown_REASON
   - DEATH_OTHER
   - unknown

Done. Processed 55 columns:

Number of type ENUMERATED: 38 with 189 distinct values
Number of type INTEGER: 11
Number of type FLOAT: 0
Number of type BOOLEAN: 0
Number of type STRING: 3
Number of type DATE: 2
Number of type DATETIME: 0

Redundant columns: []
```

Listing 2: Report file from the ETLHelper program.

```
Welcome to the BBMRI-ERIC ETL Helper Utility 1.0!

Transforming data to comply with the definition of namespace "ccdg".

=== Uploading files into database ===
Uploading data (data.tsv)... OK
Uploading source namespace (local.tsv)... OK
Uploading target namespace (ccdg.tsv)... OK
Uploading matching (local-ccdg.tsv)... OK
Uploading files into database completed.

========= SUMMARY OF INPUT DATA =========

=== Summary of input data (from Excel/CSV file) ===
Number of patients:           1066
Number of data records:       50020   (Lines in the CSV file or non-empty cells in the Excel file.)
Number of different concepts: 53   (Number of columns in the Excel file or distinct values in the CSV file's attribute
column.)
Number of empty concepts:     0   (Number of data records in the CSV/Excel file without a concept (SOURCE_SLOT), only
applicable if the CSV/MDR route is taken.)

=== Summary of the source namespace ===
Number of different data elements: 1   (Number of distinct MDR URNs; only applicable if the CSV/MDR route is taken.)
Number of different SOURCE_SLOTs: 53   (Number of distinct references to the CSV file's attribute column entries; only
applicable if the CSV/MDR route is taken.)
Number of empty SOURCE_SLOTs:     0   (If > 0, this indicates that the MDR's metadata is missing the required SOURCE slot
entries; only applicable if the CSV/MDR route is taken.)

=== Summary of the target namespace ===
Number of different data elements: 56   (Number of distinct MDR URNs.)

=== Summary of the mapping rules ===
Number of different source strings (data element and value): 205   (Concept path including the value for value sets and
Boolean, or the data type for anything except value sets or Boolean.)
Number of found mappings between source and target strings: 193   (Should be the same value as above; if not, this indi-
cates missing mappings.)

========= COMPILING INFORMATION INTO SINGLE TABLE =========

=== Performing actions on data tables ===
Building translation table... OK
Merging tables into single data table... OK
Performing actions on data tables successful!

=== Assessment of mapping completeness ===
Number of data records (facts) that should have a mapping: 50020
Number of data records (facts) that do have a mapping: 49263
Number of data records (facts) that don't have a mapping: 757

  => 98.49% of the data records (facts) have a mapping.

  Items (not facts) without a mapping (consider updating the mapping file):

         TARGETED_THERAPY_SCHEME = 2
         TARGETED_THERAPY_SCHEME = Avastin
         TARGETED_THERAPY_SCHEME = Erbitux
         TARGETED_THERAPY_SCHEME = IGN-CN
         TARGETED_THERAPY_SCHEME = Panorex
         TARGETED_THERAPY_SCHEME = Rituximab
         TARGETED_THERAPY_SCHEME = Tivozanib
         TARGETED_THERAPY_SCHEME = Vectibix
         TARGETED_THERAPY_SCHEME = unknown
         THERAPY_RESPONSE = 0
         UICC_STAGE = Not known
         WHO_GRADE = Not known

  Items (not facts) for which no data was processed (because of missing mapping or missing data in the source table):

         TARGETED_THERAPY_SCHEME = 2
         TARGETED_THERAPY_SCHEME = Avastin
         TARGETED_THERAPY_SCHEME = Erbitux
         TARGETED_THERAPY_SCHEME = IGN-CN
         TARGETED_THERAPY_SCHEME = Panorex
         TARGETED_THERAPY_SCHEME = Rituximab
         TARGETED_THERAPY_SCHEME = Tivozanib
         TARGETED_THERAPY_SCHEME = Vectibix
```

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.

```
         TARGETED_THERAPY_SCHEME = unknown
         THERAPY_RESPONSE = 0
         UICC_STAGE = Not known
         WHO_GRADE = Not known

========= FACTS DATA TRANSFORMATION PROCESS =========

=== Performing actions on data tables ===
Performing datatype transformations (casting) ... OK

  The following data type castings (from TYPE => TYPE) were applied:

         DATE => DATE: 2132 (ERROR: 0; WARNING: 0; OK: 2132)
         ENUMERATED => BOOLEAN: 1119 (ERROR: 0; WARNING: 0; OK: 1119)
         ENUMERATED => ENUMERATED: 35786 (ERROR: 0; WARNING: 0; OK: 35786)
         INTEGER => INTEGER: 7572 (ERROR: 479; WARNING: 0; OK: 7093)
         STRING => STRING: 2654 (ERROR: 0; WARNING: 0; OK: 2654)

  The following data type castings triggered errors:

         INTEGER => INTEGER: RADIATION_THERAPY_END_RELATIVE = unknown => Radiation therapy / Date of end of radiation
therapy = INTEGER
         INTEGER => INTEGER: PHARMACOTHERAPY_END_RELATIVE = unknown => Pharmacotherapy / Date of end of pharamcotherapy =
INTEGER
         INTEGER => INTEGER: SURGERY_START_RELATIVE = unknown => Surgery / Time difference between initial diagnosis and
surgery = INTEGER
         INTEGER => INTEGER: THERAPY_RESPONSE_TIMESTAMP_RELATIVE = unknown => Response to thearapy / Date response was
obtained in weeks since initial diagnosis = INTEGER
         INTEGER => INTEGER: PHARMACOTHERAPY_START_RELATIVE = unknown => Pharmacotherapy / Date of start of pharama-
cotherapy = INTEGER
         INTEGER => INTEGER: TARGETED_THERAPY_START_RELATIVE = unknown => Targeted therapy / Date of start of targeted
therapy = INTEGER
         INTEGER => INTEGER: RADIATION_THERAPY_START_RELATIVE = unknown => Radiation therapy / Date of start of radiation
therapy = INTEGER
         INTEGER => INTEGER: TARGETED_THERAPY_END_RELATIVE = unknown => Targeted therapy / Date of end of targeted thera-
py = INTEGER

=== Information about the used transformation rules ===
Number of transformation rules used: 49263
Number of warnings: 0
Number of errors: 479
Number of good transformations: 48784

  => 99.03% of the data records that do have a mapping could be transformed.

=== Final summary about the ETL process ===

In total, 97.53% of the input data records could be mapped and transformed.

========= XML CREATION PROCESS =========

Pulling data element information from REST service ... URL is: https://ccdc.bbmri-eric.eu/osseimport/mdrkeylist
... Done!

The XML file local-ccdg.xml has been created successfully in the directory ./xmlFiles/.

If you experience any problems with this program, please contact christian.knell@xxx.de or sebastian.mate@xxx.de.
```

Horizon 2020

ADOPT BBMRI-ERIC is funded by the European Union (EU)
Horizon 2020 under Grant number 676550.