GigaScience, 10, 2021, 1–13

https://doi.org/10.1093/gigascience/giab068
Technical Note

TECHNICAL NOTE

A scalable software solution for anonymizing high-dimensional biomedical data

Thierry Meurers [1,*], Raffael Bild [2], Kieu-Mi Do [3] and Fabian Prasser [1]

[1]Berlin Institute of Health at Charité–Universitätsmedizin Berlin, Medical Informatics, Charitéplatz 1, 10117 Berlin, Germany; [2]School of Medicine, Technical University of Munich, Ismaninger Str. 22, 81675 Munich, Germany and [3]Faculty of Informatics, Technical University of Munich, Boltzmannstr. 3, 85748 Garching, Germany

*Correspondence address. Thierry Meurers, Berlin Institute of Health at Charité–Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany. E-mail: thierry.meurers@charite.de https://orcid.org/0000-0001-8168-7067

Abstract

Background: Data anonymization is an important building block for ensuring privacy and fosters the reuse of data. However, transforming the data in a way that preserves the privacy of subjects while maintaining a high degree of data quality is challenging and particularly difficult when processing complex datasets that contain a high number of attributes. In this article we present how we extended the open source software ARX to improve its support for high-dimensional, biomedical datasets. **Findings:** For improving ARX's capability to find optimal transformations when processing high-dimensional data, we implement 2 novel search algorithms. The first is a greedy top-down approach and is oriented on a formally implemented bottom-up search. The second is based on a genetic algorithm. We evaluated the algorithms with different datasets, transformation methods, and privacy models. The novel algorithms mostly outperformed the previously implemented bottom-up search. In addition, we extended the GUI to provide a high degree of usability and performance when working with high-dimensional datasets. **Conclusion:** With our additions we have significantly enhanced ARX's ability to handle high-dimensional data in terms of processing performance as well as usability and thus can further facilitate data sharing.

Keywords: data privacy; anonymization; de-identification; heuristics; genetic algorithm; software tool; privacy preserving data publishing; biomedical data; data protection

Introduction

Big data technologies and the latest data science methods promise to be valuable tools for providing new insights into the development and course of diseases. These insights can be used to derive new preventive, diagnostic, and therapeutic measures [1]. Implementing these methods in practice requires access to comprehensive, multi-level datasets of high quality. At a large scale, this can only be achieved by fostering the reuse of data from different contexts and the sharing of data across institutional boundaries. The reuse of data is also in line with the FAIR (Findable, Accessible, Interoperable, Reusable) data principles and supports the reproducibility of research. However, in the context of biomedical research, sharing data is challenging because it is important to account for ethical aspects [2] and privacy concerns, as well as data protection laws, e.g., the US Health Insurance Portability and Accountability Act (HIPAA) [3] or the European General Data Protection Regulation (GDPR) [4].

One important building block for ensuring privacy is to provide safe data that minimize disclosure risks [5]. This can be achieved by using data anonymization techniques that transform the data to mitigate privacy risks [6, 7]. Typically, the anonymization process is not limited to the removal of directly identifying attributes such as the name, telephone number, or

Received: 30 September 2020; **Revised:** 19 July 2021; **Accepted:** 9 September 2021

insurance ID number. Instead, it must also account for attributes such as the postal code, age, and sex that could be combined to re-identify individuals or derive sensitive personal information [8–10]. However, transforming the data will also have an impact on its usefulness and striking the right balance between privacy and data quality is challenging. The complexity of this task is also demonstrated by several re-identification attacks [11, 12]. Because most anonymization approaches are based on the idea of reducing the uniqueness of attribute combinations, preserving a reasonable amount of information becomes particularly difficult when working with high-dimensional datasets that contain a high number of attributes [13]. Furthermore, the number of possible transformations of a dataset usually increases exponentially with the number of its attributes, leading to computational challenges [14]. Thus, the literature mostly addresses the anonymization of low-dimensional datasets featuring $\leq$10 or 15 attributes [15–18]. To put anonymization of high-dimensional datasets into practice, tools that support a variety of mathematical and statistical privacy models and allow for their combination must feature scalable algorithms capable of approximating suitable solutions. An example of such a tool is the open source software ARX [6, 19]. It is focused on biomedical data and has been mentioned in several official policies and guidelines [20, 21], used in research projects [22–24], and enabled several data publishing activities [25–27].

Versions of ARX up to 3.8.0 were only able to process datasets with a limited number of attributes that could be considered during anonymization (up to ~15). There were 2 reasons for this: (i) the software only had limited support for anonymization algorithms able to process high-dimensional data and (ii) the GUI was not designed to work with datasets containing a high number of attributes.

In this Technical Note, we describe our efforts to overcome these limitations by (i) extending ARX's user interface with additional views that simplify the management of high-dimensional data, (ii) implementing 2 novel heuristic anonymization algorithms, and (iii) evaluating the novel algorithms regarding their performance for anonymizing low-dimensional and high-dimensional datasets.

Materials and Methods

In this section, we first provide some fundamental details about data anonymization. Second, we present important properties of the ARX Anonymization Tool that had an influence on our design decisions. Third, we present the extensions implemented into ARX. Finally, we provide insights into our experimental setup.

Fundamentals of data anonymization

When anonymizing a dataset the first step is to remove all attributes that directly identify the individuals. Thereafter, the dataset is modified or noise is introduced so that the risk of identified or identifiable individuals being linked to 1 or multiple records of the dataset or to sensitive information in general is lowered [7]. This step involves the use of mathematical or statistical privacy models to quantify the risk of privacy breaches, as well as quality models that measure the usefulness of the output data. For (i) measuring privacy risks, (ii) measuring data quality, and (iii) transforming the data a variety of models can be used and combined.

Figure 1 shows a simplified example of an anonymization process. The transformation involves different procedures such as (1) randomly sampling the records, (2) aggregating values by replacing them with their mean, (3) suppressing values, (4) masking trailing characters of strings, (5) categorizing numerical values, and (6) generalizing categorical attributes. These transformations may reduce the fidelity of the data but also reduce the risk of linkage attacks and the attacker's accuracy when linking records. Furthermore, an additional uncertainty could be created by introducing noise. The transformed output data of the example fulfills 2 frequently used privacy models: k-anonymity with $k = 3$ [28] and (ε, δ)-differential privacy with $\varepsilon \approx 0.92$ and $\delta \approx 0.22$ [29].

The simple example demonstrates the variety of possibilities available for transforming data. Furthermore, it also suggests why it is often not feasible to search the entire solution space of all potential output datasets when processing more complex data. For this kind of task, solutions that try to determine a good transformation scheme on a best-effort basis, e.g., based on heuristic strategies [15, 30, 31] or clustering algorithms [16, 17, 32], have been developed. An overview of common types of approaches is provided by Fung et al. [7].

The ARX anonymization tool

ARX supports a variety of privacy models, quality models, and data transformation schemes and allows for their arbitrary combination [6]. For transforming the data, it relies on domain generalization hierarchies that describe how values can be transformed to make them less unique. For each hierarchy it is possible to define multiple levels of generalization that cover an increasing range of the attribute's domain. The basic solution space that is used by ARX is given by all possible combinations of generalization levels defined by the hierarchies. These combinations are referred to as generalization schemes.

Figure 2a shows exemplary generalization hierarchies for the attributes body mass index, sex, and ICD code. Figure 2b illustrates how the solution space resulting from these hierarchies is structured and how applying different generalization schemes would alter an exemplary dataset.

Mathematically, the solution space is a lattice [33, 34], which grows exponentially in size in accordance with the number of attributes that need to be protected [31]. As ARX is also able to apply different generalization schemes automatically to different parts of the input dataset the size of the solution space may grow further by a multiplicative factor representing the number of rows [6]. ARX supports different algorithms for finding optimal solutions within solution spaces of tractable size [35], as well as a heuristic algorithm for larger search spaces that tries to determine a good transformation scheme on a best-effort basis [31].

In addition to its anonymization engine, ARX also features a cross-platform GUI. An overview of the different perspectives provided by the platform is shown in Fig. 3.

In the "configuration" perspective it is possible to define risk thresholds for different types of attacks, to prioritize attributes by importance, to model the background knowledge of possible attackers, and to define transformation methods and rules. In the "exploration" perspective, relevant anonymization strategies are visualized for the input data and a categorization according to output data quality is supported. A further perspective supports the manual "quality analysis" of the output data. Different methods for measuring the information content of the output data, descriptive statistics, and methods for comparing the usefulness of the input and output data for different applica-

Age	Sex	ZIP	BMI	Diagnosis
55	M	92855	28.0	I47.1 Supraventricular tachycardia
91	M	94552	24.9	I50.1 Left ventricular failure
85	F	92855	26.7	I50.0 Congestive heart failure
68	F	94230	31.7	I47.1 Supraventricular tachycardia
77	F	92855	18.3	I47.0 Re-entry ventricular arrhythmia
82	F	92855	24.0	I50.0 Congestive heart failure
78	F	92301	28.1	I50.0 Left ventricular failure
75	M	94550	31.0	I47.0 Re-entry ventricular arrhythmia
82	M	94501	34.9	I47.0 Re-entry ventricular arrhythmia
69	M	93321	32.3	I50.1 Left ventricular failure

1 2 3 4 5 6

Age	Sex	ZIP	BMI	Diagnosis
81.7	F	92***	19-29	I50 Heart failure
81.7	F	92***	19-29	I50 Heart failure
81.7	F	92***	19-29	I50 Heart failure
75.0	*	94***	≥ 30	I47 Paroxysmal tachycardia
75.0	*	94***	≥ 30	I47 Paroxysmal tachycardia
75.0	*	94***	≥ 30	I47 Paroxysmal tachycardia

1 Sampling
2 Aggregation
3 Suppression
4 Masking
5 Categorization
6 Generalization

Figure 1: Exemplary anonymization process.

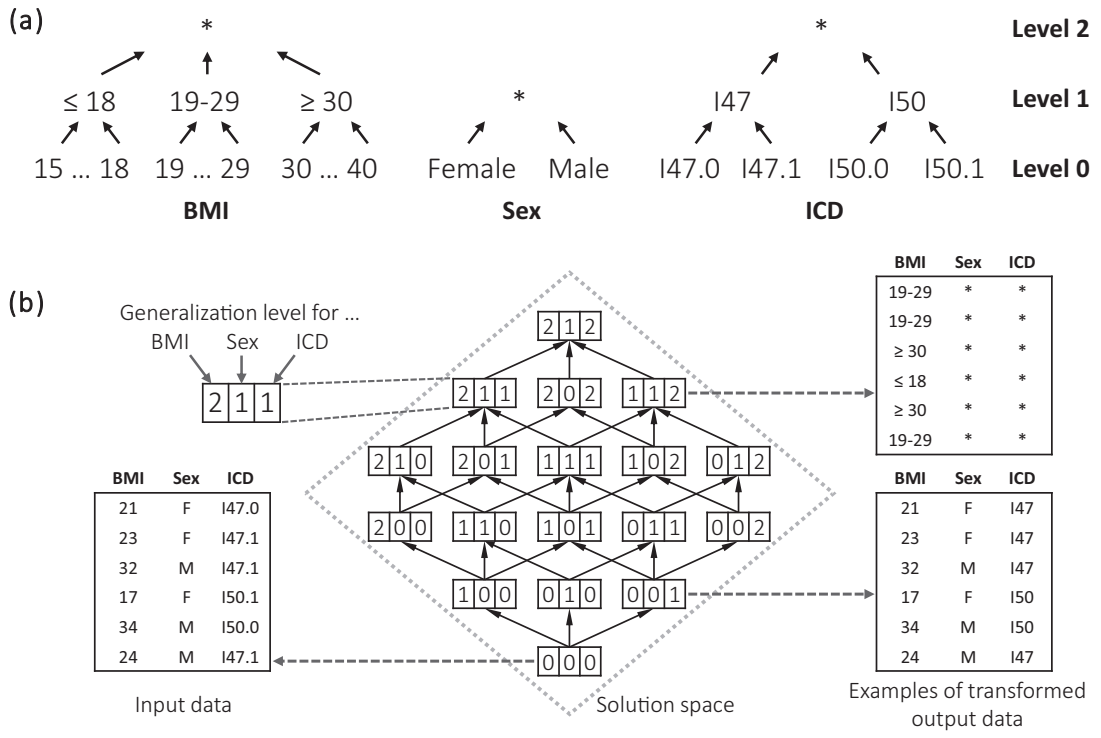

Figure 2: Generalization hierarchies (a) and the structure of the corresponding solution space together with examples of how data are transformed when applying different generalization schemes using global generalization (b).

tion scenarios are provided. In a "risk analysis" perspective, it is possible to visually compare input and output data using different risk models. However, in the user interface it is challenging to support high-dimensional datasets. For example, several perspectives and views of the software display lists of all attributes of the dataset loaded, which can become confusing and lead to performance problems on some platforms with an increasing number of attributes.

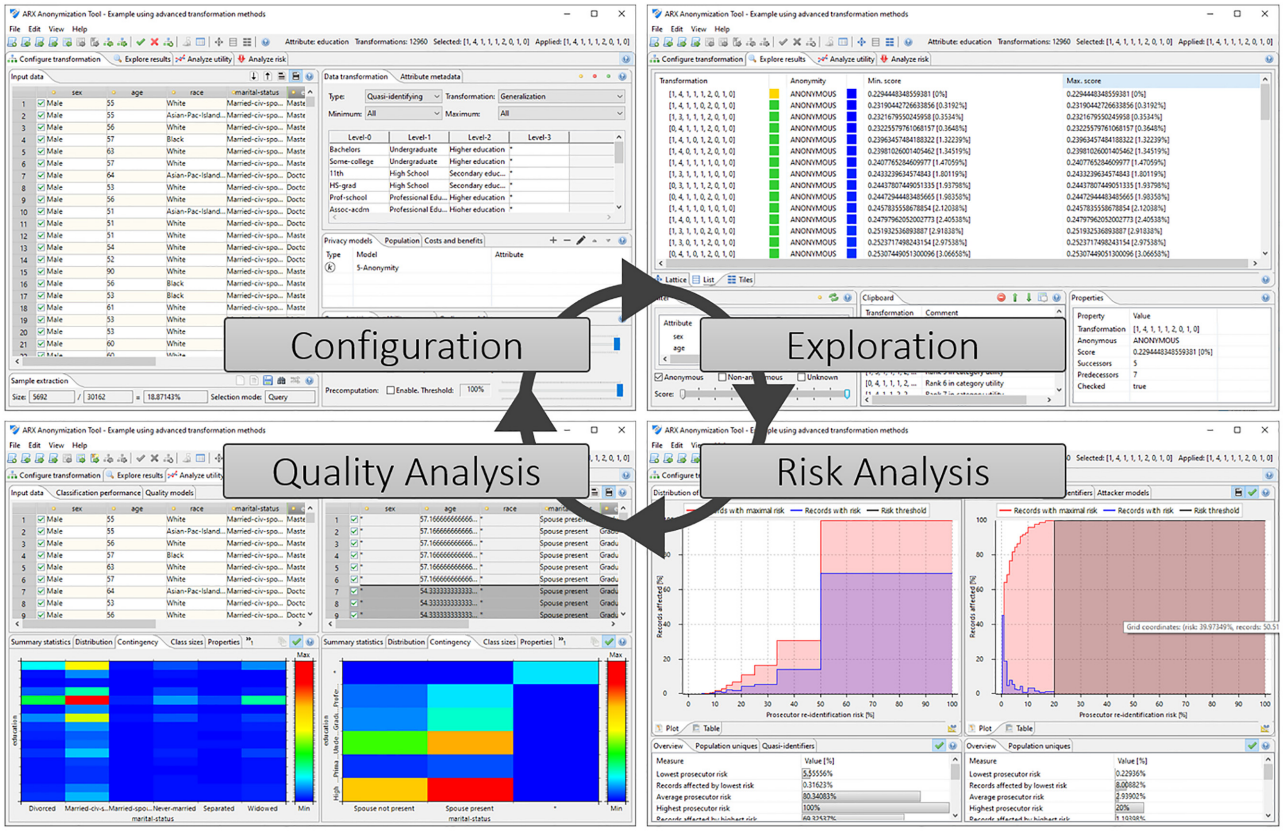

Figure 3: Basic perspectives of the graphical interface of the ARX Data Anonymization Tool.

Integrating anonymization algorithms for high-dimensional data

As mentioned above, the anonymization procedures supported by ARX are built around a basic operator that searches through the generalization lattice. In prior work we have already integrated a greedy best-first bottom-up search algorithm into the software [31]. This algorithm starts at the bottom generalization scheme, which applies no generalization to the data. It then "expands" this generalization scheme by applying all generalization schemes to the input dataset that can be derived by increasing 1 of the generalization levels. The quality of the resulting output dataset is computed for all these schemes, and the process is repeated by expanding the generalization, resulting in the dataset with highest quality. This process is then repeated until a user-specified period of time has passed. During the execution of the algorithm, a list of all generalization schemes that have been evaluated is stored and, in each iteration, the scheme with the highest output data quality that has not yet been expanded is expanded. For further details we refer readers to the original publication [31].

It must be noted that this process is only suitable for processing datasets of medium dimensionality (~15 attributes) for several reasons. First, the search process may become trapped in local minima because there is no significant diversification of the solutions considered. Second, the process naturally favors transformation schemes located in the lower part of the search space (i.e., schemes that apply a low degree of generalization). While this makes sense for anonymization processes that only apply generalization, the method reaches its limits with the complex transformation operations supported in newer versions of ARX in which different transformation schemas are used to transform different parts of a dataset. In this case, a better overall solution can sometimes be determined if outliers are transformed more strongly. To further improve this process, we have integrated 2 new algorithms for processing high-dimensional data into the software.

The first algorithm closely resembles the bottom-up greedy best-first search but performs this process top-down. We do not describe it in further detail because this is a straightforward extension of the process described in the previous paragraphs.

The second algorithm applies a genetic optimization process to the anonymization problem. Genetic algorithms search for solutions in a heuristic manner that is oriented on the process of natural selection [36]. During the search, the solutions are considered individuals that carry the solution's properties encoded as a list of genes (in our application, the individuals carry generalization schemes and genes correspond to the generalization level of specific attributes). The set of candidate solutions/individuals is called population. Mostly, the initial population is created by randomly generating individuals. Thereafter, the algorithm works iteratively. By crossing (i.e., randomly combining the properties of 2 individuals) and mutating (i.e., randomly altering the properties of individuals) selected individuals contained in the population, each iteration will result in a new, so-called, generation. Whether and how an individual is altered is determined by its fitness, which usually is calculated using the cost function of the investigated optimization problem. Once reaching a predefined limit of iterations the fittest individual is considered the optimal solution. However, there is no guarantee that a globally optimal solution can always be found.

We opted for the genetic algorithm because it is one of the most well-known population-based meta-heuristics. In comparison to single-solution-based algorithms (e.g., simulated annealing or the aforementioned greedy heuristics) population-based approaches maintain multiple candidate solutions, which potentially results in a high degree of diversification and a decreased risk of getting stuck in local optima [37]. Moreover, genetic algorithms have already been successfully applied for anonymizing data in previous work. However, prior approaches were often limited to a specific kind of data or privacy model (see Section "Comparison with Prior Work"). The genetic algorithm implemented into ARX is based on the work by Wan et al. [38]. Wan et al. used the algorithm for anonymization of genomic data using a game-theoretic privacy model, which was already successfully adapted and integrated into ARX in prior work [39].

Figure 4 illustrates how the algorithms search through the solution space to find a good generalization scheme, based on the example presented in Fig. 2. Although the process shown in the figure is simplified, it illustrates that both approaches follow completely different concepts.

To make the genetic algorithm compatible with the types of solution spaces used by ARX and to integrate it with the privacy and quality models supported by the software, every individual carries a list of numerical values representing a generalization scheme. The list's length (i.e., the number of genes) equals the number of attributes that need to be transformed and the ith value of the list represents the generalization level of the ith attribute. The range of each value is given by the lowest and highest generalization level available for the corresponding attribute. Applied to the example illustrated in Fig. 4, this results in individuals carrying 3 genes with values between 0 and 2. Implementation-wise the populations are maintained in a matrix-like structure with the rows of this matrix representing individuals (generalization scheme) and columns their genes (generalization level of an attribute). ARX's privacy and quality models have been integrated via the fitness function. ARX always automatically alters the output of any given transformation in such a way that the required privacy guarantees are provided. This is achieved by suppressing records [40]. The suppression of records is captured by a decrease in data quality. Hence, we defined the fitness of a transformation to equal output data quality, which not only measures the transformation's direct effect on data quality but also implicitly captures how well the required privacy guarantees are achieved.

The algorithm itself works as follows:

Initialization: During the initialization 2 equally sized subpopulations are created. Following the approach of Wan et al. [38], the first individuals of the first subpopulation are generated in the form of a "triangle pattern" using the lowest and highest generalization levels. An example is provided in Fig 5. The remaining individuals of the first subpopulation, as well as the entire second subpopulation, are filled by randomly creating individuals. The motivation behind this approach is based on properties of genetic data [41]. To determine whether the initialization procedure is also favorable in our case, we performed experiments in which we compared the initialization strategy proposed by Wan et al. with a completely random initialization in a single- as well as a dual-population setting (see Supplementary Table S1). The experiments showed that using the "triangle pattern" performs well when processing low-dimensional data and significantly outperforms other approaches when processing high-dimensional datasets. This can be explained by the fact that the pattern creates populations that cover a larger part of the solution space in the beginning.

Iteration: After initializing the subpopulations the algorithm's main loop is started. The algorithm stops after reaching a pre-defined number of iterations or time limit. Within the loop the following steps are executed:

Step 1: Sorting: The individuals contained in the subpopulations are sorted by their fitness in descending order.

Step 2: Selection: The fittest individuals of the current population will simply be copied to the next generation without being modified. We refer to this fraction of individuals as the "elite fraction." In Fig. 4b, this mechanism is indicated by an arrow, with which an individual points to itself because it remains unchanged.

Step 3: Crossover: Next, the so-called "crossover fraction" of the new generation is populated. For this purpose, 2 parent-individuals from the "production fraction" of the current population are crossed to generate a new child-individual. The probability of being chosen as a parent increases with the fitness. The crossover is performed in a randomized fashion. For every gene it is decided randomly from which of the 2 parents it is inherited. Figure 4b illustrates how a child-individual inherits the genes of its ancestors. Which gene was inherited from which parent can be distinguished by the color coding.

Step 4: Mutation: The rest of the new generation is populated by randomly choosing individuals of the current generation and mutating them by altering their genes. The number of changed genes is randomly chosen between 1 and an upper bound, which is calculated by multiplying the "mutation probability" by the number of available genes. Figure 4b depicts an individual that is being mutated at 1 of its genes while leaving the remaining genes unchanged. The mutated gene is indicated by a change in color.

Step 5: Swapping: Additionally, it is possible that the fittest individuals are swapped between the 2 subpopulations. How often they are changed depends on the "immigration interval," which refers to the number of iterations between the swaps. The number of exchanged individuals can be controlled by the "immigration fraction."

Extending the user interface for high-dimensional data

ARX is implemented as a cross-platform program using Java and executed on the Java Virtual Machine. The GUI is implemented using the Standard Widget Toolkit (SWT), which enables implementing native GUIs on 3 supported platforms: Windows, Linux, and MacOS.

To improve the GUI's usability when working with high-dimensional datasets we made use of 2 SWT-based components provided by the Eclipse Nebula Project [42]. The first is NatTable. Based on the idea of virtual tables it ensures that the GUI remains responsive and provides a high rendering performance when displaying large datasets. The second is Pagination Control. This component is used to display a navigation page when working with tables used to configure a potentially large number of attributes.

Additionally, ARX features a mechanism that automatically detects the type of an attribute to ease the initial import of data as well as the ability to configure multiple attributes at once. These last 2 features are also available for smaller datasets but are especially helpful when working with high-dimensional datasets.

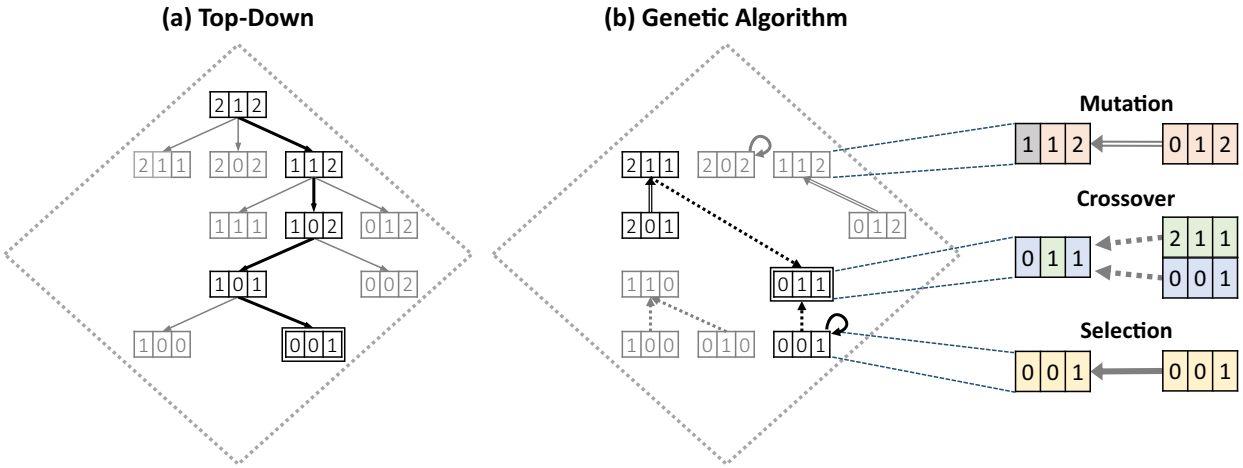

Figure 4: Illustration of (a) the top-down approach and (b) the genetic algorithm searching the solution space. Generalization schemes visited that were not on the path to the best solution are colored grey. The best scheme found is marked by a double border.

#	Individual	
1	[**3**, 0, 0, 0, 0]	Filled using a „triangle“ pattern
2	[**3**, **1**, 0, 0, 0]	
3	[**3**, **1**, **5**, 0, 0]	
4	[**3**, **1**, **5**, **3**, 0]	
5	[**3**, **1**, **5**, **3**, **1**]	
6	[2, 0, 4, 2, 1]	Randomly generated
	⋮	
n	[1, 1, 3, 2, 0]	

Figure 5: Initialization of the first subpopulation for a solution space with the highest generalization levels of [3, 1, 5, 3, 1].

Experimental design

Experiments

With the extensions described in this article, ARX now supports 3 algorithms for anonymizing high-dimensional data: (1) the initial bottom-up search, (2) the new top-down search, and (3) the new genetic search algorithm. We performed a series of experiments to study how well these algorithms work for different types of data to provide users with insights into which algorithm should be used in which context. For the experiments we used low-dimensional datasets with <10 attributes and high-dimensional datasets containing ≤30 attributes (more details about the datasets are provided in the Section “Datasets”). Depending on the dimensionality of the datasets we conducted 2 types of experiments:

(1) Experiments with low-dimensional data: We compared the algorithms to the optimal algorithm already supported by ARX [35] in the low-dimensional setting. We did this for 2 reasons. First, heuristic algorithms might also be relevant when anonymizing low-dimensional data if they significantly outperform optimal algorithms in terms of the time needed to find the optimal solution. Second, experiments with low-dimensional data might provide insights into basic strengths and weaknesses of the approaches. To this end, we compared the overall execution time of ARX’s optimal algorithm with the time needed by the heuristic algorithms to find the optimal solution.

(2) Experiments with high-dimensional data: Here, we use the 3 heuristic algorithms to anonymize high-dimensional datasets. This experiment was performed to determine whether the novel approaches (genetic and top-down) offer an advantage over the bottom-up algorithm. To this end, we executed the algorithms with different time limits and compared the quality of their results.

Privacy, quality, and transformation model

To investigate a broad spectrum of anonymization problems, we decided to use different privacy and data transformation models.

For measuring and managing privacy risks, we used 2 models:

(1) Distinguishability: To implement restrictions on the distinguishability of data, we used the well-known and relatively strict k-anonymity model. A dataset is k-anonymous if every record cannot be distinguished from $\geq k-1$ other records in respect to attributes that may be used to de-anonymize the data [43]. As a parameter we used $k = 5$, which is a common recommendation [44].

(2) Population uniqueness: ARX also supports statistical models that estimate disclosure risks by estimating the fraction of records in a dataset that are expected to be unique in the overall population. Compared to k-anonymity, this is a relatively weak privacy model. For our experiments we enforced a uniqueness of 1% within the US population and relied on the model introduced by Pitman to estimate population characteristics [45, 46].

For transforming data, we also used 2 common models:

(1) Global generalization: With this model, the values in a dataset are generalized on the basis of user-defined hierarchies. In this process, it is guaranteed that all values of an attribute undergo generalization to the same level of the associated hierarchy. To prevent overgeneralization, records can also be removed from the dataset.

(2) Local generalization: With this model, data are also transformed by generalization, but values of the same attribute in different records can be transformed differently. Records may also be removed, but this is typically not required owing to the flexibility of the transformation model.

In ARX, local transformations are implemented by using an iterative process in which the dataset is automatically partitioned and different transformation schemes are applied to different partitions [6]. In our experiments with local generalization, we used 100 iterations and different time limits for individual iterations.

To quantify data quality, we decided to use the intuitive "Granularity" model [47], which measures the value-level precision of the output data. The measurements are normalized with 0% representing a dataset from which all information has been removed and 100% corresponding to a completely unmodified dataset [6].

Datasets

For evaluating the performance of the heuristic algorithms, we used 6 different real-world datasets. An overview of the properties of the datasets is shown in Table 1.

Most of the datasets have already been used in previous evaluations of data anonymization algorithms. As low-dimensional datasets we choose (1) an excerpt of the 1994 US census dataset (Census income), which can be considered the de facto standard for evaluating anonymization algorithms; (2) data from a nationally representative US time diary survey; and (3) results from the integrated health interview series collecting data on the health of the US population.

As high-dimensional datasets we included (1) data from the responses to the American Community Survey (ACS), which captures demographic, social, and economic characteristics of people living in the United States; (2) a credit card client dataset from Taiwan used to estimate customers' default payments; and (3) answers to a psychological test designed to measure an individual's Machiavellianism from the open source psychometrics project. As attributes that needed to be transformed, we selected variables that are typically associated with a high risk of re-identification. These included demographic data, timestamps, spatial information, medical attributes, and payment histories.

Parameterization

While the top-down and bottom-up search algorithms do not require any additional parameterization, the genetic search algorithm features multiple configuration parameters, which are shown in Table 2. In ARX, these parameters are presented as configuration options to the users.

Table 2 also shows the parameters used in our evaluation. Regarding all but 1 parameter we followed the suggested configuration by Wan et al. [38] for all parameters that are applicable to our setting. We made this decision on the basis of a set of experiments performed in preparation of our evaluation in which we individually altered all parameters and examined their effect on the performance of the algorithm (see Supplementary Table S2). This process showed that setting the production fraction to 0.2 (instead of 0.8 as suggested by Wan et al.) improves execution times when processing low-dimensional datasets and data utility when processing high-dimensional datasets. The fact that almost the same parameters work well in our setting as well as in the experiments by Wan et al., although very different anonymization procedures are being investigated, can be seen as an indicator of the robustness and generality of this parameterization.

Technical Set-up

We repeated each experiment 5 times and report the average for 2 reasons: first, it is well known that execution times of JVM-based programs vary slightly owing to effects from functionalities, such as just-in-time compilation. Second, the genetic algorithm is randomized and hence may perform slightly differently in each execution.

The experiments were performed on a desktop computer with an AMD Ryzen 2700X processor (8 cores, 3.7-4.3 GHz) running 64-bit Windows 10 (version 1909) and a 64-bit Oracle JVM (version 1.8.0).

Results

Experimental results

Low-dimensional data

The results of the first set of experiments are displayed in Fig. 6. For each heuristic algorithm, it shows the time in seconds needed to determine the optimal solution (and the overall execution time for the optimal algorithm) using the global transformation model. We did not use the local transformation model in this experiment because the underlying algorithm is heuristic in nature (independently of the actual search strategy used) and therefore cannot be used to compare the time needed to achieve a specific result in terms of output data quality [6].

As can be seen, heuristic approaches provided a valuable alternative to the optimal approach even in low-dimensional settings. When aiming for a threshold on distinguishability, the bottom-up and top-down search algorithms almost always outperformed the optimal algorithm. On average, the genetic algorithm was slower than the other heuristic approaches because it aims at diversifying the solutions considered, which is not a desirable feature in low-dimensional settings. Whether the top-down approach or the bottom-up approach performed better was associated with the degree of generalization required and hence with the fact whether the optimal solution is located closer to the top or to the bottom of the lattice.

When optimizing for a threshold on population uniqueness the optimal algorithm outperformed the heuristic approaches in 2 of 3 cases. This can be explained by the fact that calculating population uniqueness is much more computationally complex than checking for k-anonymity because bivariate non-linear equation systems need to be solved. As a consequence, execution times are not dominated by the time needed to transform the dataset but by the time needed to evaluate the privacy model. The optimal approach implements a wide variety of pruning strategies that reduce the number of transformations that need to be checked [40], which cannot be implemented by the heuristic algorithms. The genetic algorithm provided the worst overall performance because it tries to look at a diverse set of potential solutions.

High-dimensional data

The results of the experiments with high-dimensional data are displayed in Figs 7 and 8. We compared the development of output data quality for the different algorithms over time and present 2 different types of results. For global transformation we continuously measured the development of output data quality over time. For local transformation we present the output data quality achieved with different time limits because the heuristic nature of the local transformation algorithm implemented in ARX makes it difficult to directly track the progress [6].

Figure 7 shows the development of output data quality over time when using the global transformation model until the results of all 3 algorithms stabilized. As can be seen, all algorithms almost always eventually found a solution with comparable quality. However, when enforcing a threshold on popula-

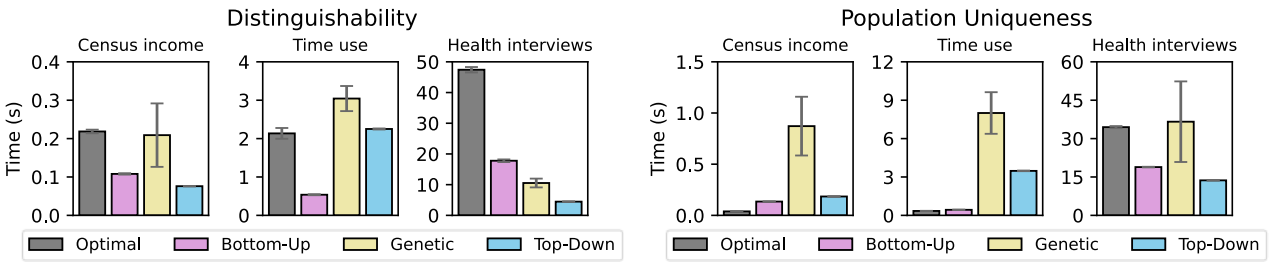

Figure 6: Time required to find an optimal solution for different low-dimensional datasets and privacy models depending on the algorithm used.

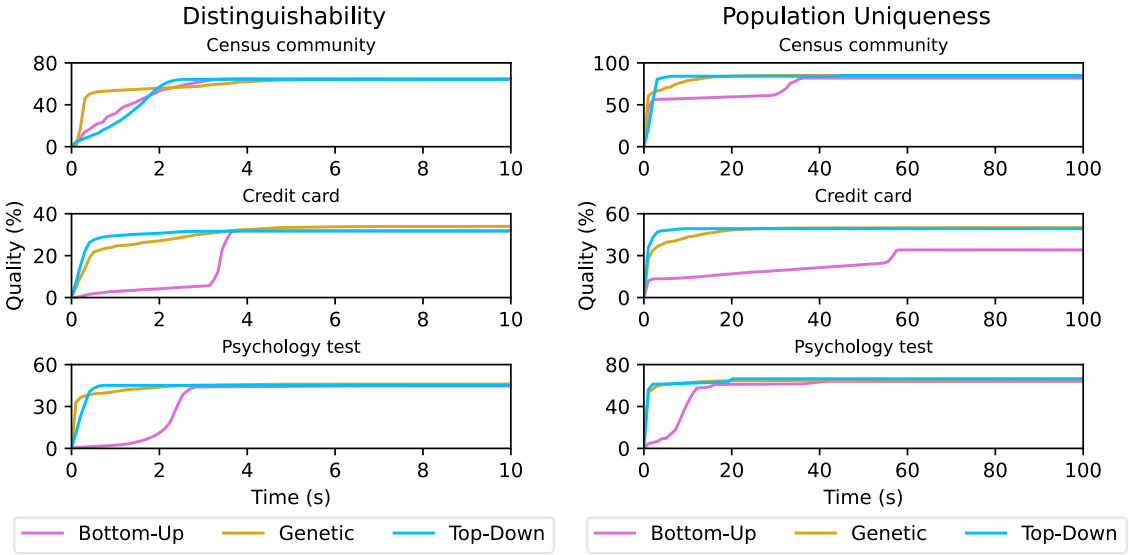

Figure 7: Global generalization: Quality improvement over time for different high-dimensional datasets and privacy models depending on the algorithm used.

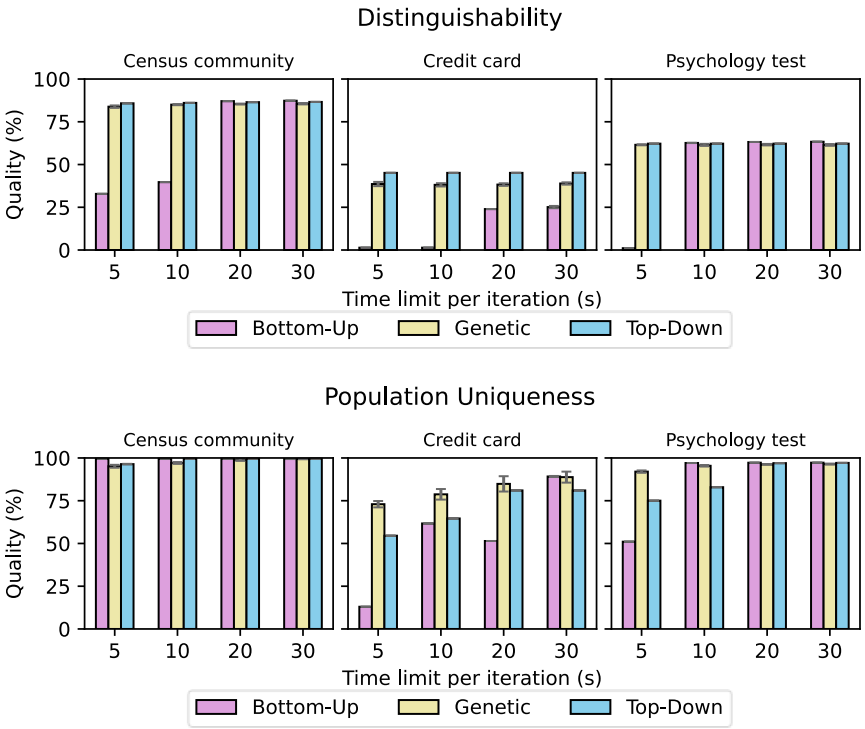

Figure 8: Local generalization: Quality achieved for different high-dimensional datasets and privacy models depending on the algorithm used.

Table 1: Overview of the datasets used for comparing the algorithms

Name	No. Attributes	No. Records	Solution space size	Category
Census income [48]	9	30,162	12,960	Low dimensional
Time use [49]	9	539,254	34,992	Low dimensional
Health interviews [50]	9	1,185,424	25,920	Low dimensional
Census community [51]	30	68,725	203,843,174,400	High dimensional
Credit card [52]	24	30,000	49,478,023,249,920	High dimensional
Psychology test [53]	16	73,489	85,030,560	High dimensional

Table 2: Parameters of the genetic algorithm and the values used in the experiments

Parameter	Description	Value
Elite fraction	Fraction of individuals that is directly copied to the next generation	0.2
Crossover fraction	Fraction of individuals that is replaced by new individuals that are generated by crossing 2 parents from the production fraction	0.4
Production fraction	Fraction of individuals used as parents when generating crossover individuals	0.2
Mutation probability	Used to calculate the upper bound of changed genes when mutating individuals	0.05
Immigration fraction	Fraction of individuals that is swapped between the subpopulations	0.2
Immigration Interval	No. of iterations between swaps	10
Iterations	No. of iterations performed by the genetic algorithm	50
Subpopulation size	No. of individuals contained in each of the subpopulations	50

tion uniqueness on the credit card dataset, the bottom-up algorithm exhibited suboptimal performance. Moreover, in most cases the genetic and top-down approach found better solutions much quicker than the bottom-up algorithm. When comparing the different algorithms to each other it can be seen that the genetic algorithm was generally good at quickly determining a relatively good solution while the top-down algorithm provided a good balance of optimization speed and quality of its overall output. It can also be seen that output data quality was higher when reducing population uniqueness compared to reducing distinguishability, as the former model is weaker than the latter (see Section "Privacy, quality, and transformation model").

Figure 8 provides additional insights by presenting the results for the local transformation model.

Again, the time axis covers the time that was needed for the solutions of the different algorithms to stabilize. As can be seen, the results are quite similar to the results obtained using the global transformation model, apart from the fact that the overall output data quality is higher with this transformation method. The genetic algorithm is good at very quickly finding a relatively good transformation, and in most cases all algorithms finally found a comparable solution. The credit card dataset is a notable exception. In this case, the bottom-up algorithm provided the best result when reducing population uniqueness and the top-down approach provided the best result when reducing distinguishability. It is notable that the genetic algorithm performed best for short time limits in the former case because the credit card dataset results in the largest solution space and the evaluation of individual solution candidates is expensive for population uniqueness. Moreover, good solutions were not located close to the top or bottom of the search space. This is exactly the scenario in which one would expect good performance from a genetic search process.

Extended user interface

In the updated version of the ARX GUI, 7 views of the software distributed over all 4 perspectives have been extended using the pagination feature. We note that this extension is graceful, meaning that it is only activated when a high-dimensional dataset is loaded into the software (an appropriate threshold can be specified in the tool's settings). As an example, the pagination feature of a view in ARX's quality analysis perspective is shown in Fig. 9.

Further features that are important for managing high-dimensional data with ARX, such as auto-detection of data types and options to configure multiple attributes at once, are located in different parts of the GUI, such as data import and hierarchy creation wizards, as well as the software's main toolbar.

Discussion

Principal results

In this article we have presented the results of our efforts to improve the ability of the ARX Anonymization Tool to handle high-dimensional data. For this purpose, we extended the GUI and introduced and evaluated 2 new heuristic anonymization algorithms. One of the algorithms, top-down search, complements the existing greedy bottom-up search algorithm. The other approach is based on a genetic algorithm and aims at diversifying the potential solutions considered using the process of natural selection.

Evaluating the newly implemented algorithms showed that they are particularly useful in scenarios where high-dimensional data need to be anonymized. Using global generalization, they clearly outperformed the previously implemented bottom-up search (i.e., better performance in 5 of the 6 experiments). A similar result was observed when using local generalization. Averaged over all experiments, the new algorithms achieved a utility of 76.5% (genetic algorithm) and 75.1% (top-down algorithm), which is significantly higher than that provided by the bottom-up approach (60.2%). Especially when anonymizing the dataset with the largest solution space (credit card), the new algorithms often performed significantly better, in terms of both scalability and utility. Additionally, the results obtained when processing low-dimensional data showed that

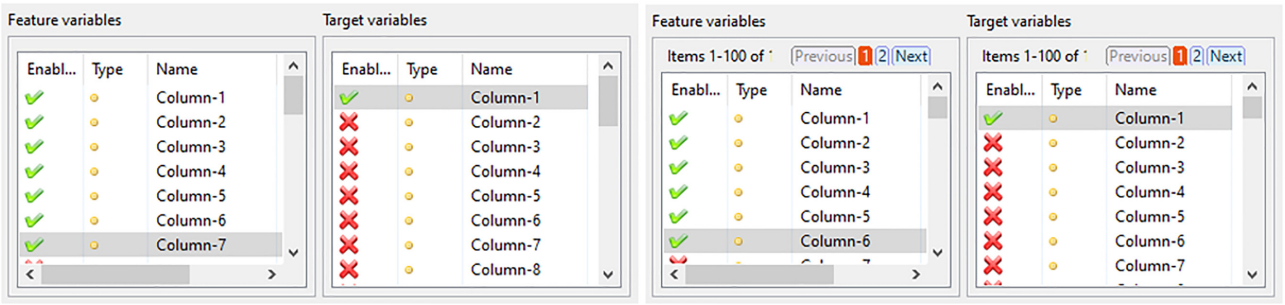

Figure 9: Screenshots from the "Classification model" tab before (left) and after (right) adding the pagination feature.

heuristic algorithms can be helpful to improve computational efficiency even in scenarios where optimal algorithms could be used. The top-down approach required the least amount of time on average to find an optimal solution (4.0 s), followed by the bottom-up approach (6.3 s), the genetic algorithm (9.9 s), and the optimal search strategy (14.1 s).

Making a general recommendation for one of the algorithms is difficult on the basis of the results of our experiments. To help users to decide on an algorithm, ARX automatically determines whether it is feasible to calculate an optimal solution or whether a heuristic algorithm should be used. Also, ARX provides the means to easily try out different algorithms and compare their results to enable users to determine which approach works best in which specific context.

Limitations

Our results show that the performance of the algorithms studied strongly depends on the dataset anonymized and the configuration used. While the new heuristic algorithms typically exhibited significantly improved performance in comparison to the methods previously implemented in ARX, this is not guaranteed to always be the case.

The exact operations of the genetic algorithm can be optimized by adjusting its parameterization. In our experiments, we used the parameterization by Wan et al. [38] and additionally tuned the parameters for optimal average performance. Therefore, we chose a single parameterization in all our experiments. Optimizing the parameters for specific use cases could therefore lead to further improvements. For this reason, the GUI and API of ARX allow the user to easily change the parameterization of the genetic algorithm.

Comparison with prior work

It has been demonstrated multiple times that genetic algorithms can be used for anonymizing data. However, previously described solutions were mostly tailored towards specific types of data or privacy and transformation models.

Examples of approaches that focus on a specific type of data include the algorithm by Wan et al. [38], which targets genomic data and which we have adopted to general tabular data in this work, and the approach for anonymizing graphs presented by Casas-Roma et al. [54].

Regarding specific privacy and transformation models, genetic algorithms have also been used in clustering-based anonymization processes. To reduce distinguishability, such algorithms partition the records of a dataset into several groups with each of the groups containing at least k members, hence implementing the k-anonymity model. Solanas et al. [55] demonstrated how the computationally challenging partition step, which aims at maximizing homogeneity within the groups, can be performed using a genetic algorithm. In their approach, the number of genes equals the number of records in the dataset, with the ith gene representing the group of the ith record. The groups are encoded as an alphabet with a fixed size as the maximal number of different groups can be derived from k and the number of records in the dataset. Lin et al. [32] described how the scalability of the clustering process can be improved for large datasets by encoding the solution using the entire population instead of a single individual. Finally, focusing on data transformations, Iyengar [47] has demonstrated how a genetic algorithm can be used to determine intervals for generalizing values. In simplified terms each individual is a binary string with a length derived from the number of processed attributes and the number of their distinct values. A value of 1 in the string implies that a value is used as an interval boundary.

Our work is different from these approaches because it integrates a genetic algorithm into ARX in such a way that it can be used to anonymize datasets using a variety of privacy models, quality models, and data transformation schemes.

Heuristics anonymization algorithms comparable to the bottom-up approach evaluated in our article include DataFly [30] and iGreedy [15]. Both use global generalization and are focused on k-anonymity only. They are based on a bottom-up search and follow the concept of minimal anonymization, meaning that they terminate as soon as they find a transformation that fulfills the requested privacy properties. In previous work we have already shown that the bottom-up algorithm implemented by ARX outperforms these approaches [31]. Furthermore, other researchers have focused on top-down search strategies. Important examples include the work of He and Naughton [56], who proposed a greedy top-down algorithm to partition a dataset and apply local generalization, as well as the Top-Down Specialization method described by Fung et al. that iteratively specializes attributes until violating the anonymity requirements [57].

Conclusion and Future Work

With the work presented in this article we have significantly enhanced ARX's ability to handle high-dimensional data, both in the GUI and the API. All features described in this article are available as open source software and are included in the latest release of the software [19].

In future work, we plan to add additional features to improve ARX's performance for high-dimensional data. While ARX already supports a wide range of data transformation models, we believe that the addition of further transformation methods would have the largest impact. One important example

is sub-tree generalization, which provides a good balance between improved output data quality and interpretability of output datasets [58]. Moreover, we plan to add further methods from the area of statistical disclosure control, such as Post-Randomization (PRAM), that can be used to inject uncertainty into data with little impact on its usefulness [59].

Availability of Supporting Source Code and Requirements

Project name: ARX Anonymization Tool
Project home page: https://arx.deidentifier.org/
GitHub repository: https://github.com/arx-deidentifier/arx
Operating system(s): Platform independent
Programming language: Java 8
Other requirements: None
License: Apache License 2.0
RRID:SCR_021189
bio.tools ID: arx
Project name: Benchmark of ARX's Heuristic Algorithms
GitHub repository: https://github.com/arx-deidentifier/genetic-benchmark
Operating system(s): Platform independent
Programming language: Java 8, Python 3
Other requirements: None
License: Apache License 2.0

Data Availability

The datasets used to benchmark the algorithms are publicly available. The corresponding download URLs are referenced in Table 1. Additionally, the datasets are part of a GitHub repository containing our benchmarking code [60]. The repository also contains the generalization hierarchies used for anonymizing the data and the raw benchmark results as .csv files. A snapshot of the code and the supporting data is available in the *GigaScience* GigaDB repository [61].

Abbreviations

API: Application Programming Interface; GUI: graphical user interface; ICD: International Classification of Diseases.

Competing Interests

The authors declare that they have no competing interests.

Authors' Contributions

R.B. initiated and conceptualized the work. F.P., K.D., and T.M. implemented the novel anonymization algorithms and integrated them into ARX. F.P. and T.M. reworked the user interface of ARX. T.M. programmed the framework used to evaluate the novel algorithms and performed the benchmarks. T.M. and F.P. drafted the manuscript. R.B. and K.D. revised the manuscript and provided important suggestions for further improvements. All authors read and approved the final manuscript.

References

1. Schneeweiss S. Learning from big health care data. N Engl J Med 2014;**370**(23):2161–3.
2. Ballantyne A. Where is the human in the data? A guide to ethical data use. Gigascience 2018;**7**(7):doi:10.1093/gigascience/giy076.
3. Office for Civil Rights, HHS. Standards for privacy of individually identifiable health information. Final rule. Fed Regist 2002;**67**(157):53181–273.
4. Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC (General Data Protection Regulation).
5. Ritchie F. Five Safes: designing data access for research. Economics Working Paper Series 1601. Bristol: University of the West of England; 2016:doi:10.13140/RG.2.1.3661.1604.
6. Prasser F, Eicher J, Spengler H, et al. Flexible data anonymization using ARX—Current status and challenges ahead. Softw Pract Exp 2020;**50**(7):1277–304.10.1002/spe.2812
7. Fung B, Wang K, Fu AW-C, et al. Introduction to privacy-preserving data publishing: Concepts and techniques. Chapman and Hall/CRC; 2010:341.
8. Rocher L, Hendrickx JM, de Montjoye Y-A. Estimating the success of re-identifications in incomplete datasets using generative models. Nat Commun 2019;**10**(1):3069.
9. Sweeney L. Simple demographics often identify people uniquely. Carnegie Mellon University, Data Privacy, 2000. http://dataprivacylab.org/projects/identifiability. Accessed 9 July 2020.
10. Majeed A, Lee S. Anonymization techniques for privacy preserving data publishing: A comprehensive survey. IEEE Access 2021;**9**:8512–45.
11. El Emam K, Jonker E, Arbuckle L, et al.. A systematic review of re-identification attacks on health data. PLoS One 2011;**6**(12):e28071.
12. Henriksen-Bulmer J, Jeary S. Re-identification attacks—A systematic literature review. Int J Inf Manage 2016;**36**(6):1184–92.
13. Aggarwal C. On *k*-anonymity and the curse of dimensionality. In: Proc. 31st International Conference on Very Large Data Bases, Trondheim, Norway. VLDB Endowment; 2005:901–9.
14. Prasser F, Kohlmayer F, Kuhn KA. Efficient and effective pruning strategies for health data de-identification. BMC Med Inf Decis Making 2016;**16**(1):49.
15. Babu K, Ranabothu N, Kumar N, et al. Achieving k-anonymity using improved greedy heuristics for very large relational databases. Trans Data Priv 2013;**6**:1–17.
16. Byun J-W, Kamra A, Bertino E, et al.. Efficient k-anonymization using clustering techniques. In: Kotagiri R, Krishna PR, Mohania M , et al., eds. Advances in Databases: Concepts, Systems and Applications. Berlin, Heidelberg: Springer; 2007:188–200.
17. Loukides G, Shao J. Clustering-based K-anonymisation algorithms. In: Wagner R, Revell N, Pernul G , eds. Database and Expert Systems Applications. Berlin, Heidelberg: Springer; 2007:761–71.
18. Lee H, Kim S, Kim JW, et al.. Utility-preserving anonymization for health data publishing. BMC Med Inf Decis Making 2017;**17**(1):104.
19. ARX Project. ARX Data Anonymization Tool. (Version 3.9.0). 2021. https://github.com/arx-deidentifier/arx. Accessed 5 July 2021.
20. External guidance on the implementation of the European Medicines Agency policy on the publication of clinical data for medicinal products for human use (EMA/90915/2016 Version 1.4). European Medicines Agency;

2018. https://www.ema.europa.eu/en/human-regulatory/marketing-authorisation/clinical-data-publication/support-industry/external-guidance-implementation-european-medicines-agency-policy-publication-clinical-data. Accessed 17 July 2020.
21. Elliot M, Mackey E, O'Hara K, et al. The Anonymisation Decision-Making Framework. UK Anonymisation Network; 2016.
22. Xu L, Jiang C, Chen Y, et al. Privacy or utility in data collection? A contract theoretic approach. IEEE J Sel Top Signal Process 2015;**9**(7):1256–69.
23. Kim J, Ha H, Chun B-G, et al. Collaborative analytics for data silos. In: 2016 IEEE 32nd International Conference on Data Engineering (ICDE), Helsinki, Finland. 2016:743–54.
24. Costa C, Chatzimilioudis G, Zeinalipour-Yazti D, et al. Efficient exploration of telco big data with compression and decaying. In: 2017 IEEE 33rd International Conference on Data Engineering (ICDE), San Diego. 2017:1332–43.
25. Kuzilek J, Hlosta M, Zdrahal Z. Open University Learning Analytics dataset. Sci Data 2017;**4**(1):doi:10.1038/sdata.2017.171.
26. Ursin G, Sen S, Mottu J-M, et al. Protecting privacy in large datasets—First we assess the risk; then we fuzzy the data. Cancer Epidemiol Biomarkers Prev 2017;**26**(8):1219–24.
27. Lean European open survey on SARS-CoV-2 infected patients - studying SARS-CoV-2 collectively. Lean European Open Survey on SARS-CoV-2 Infected Patients. https://leoss.net/. Accessed 9 July 2020.
28. Sweeney L. Achieving k-anonymity privacy protection using generalization and suppression. Int J Uncertain Fuzziness Knowl Based Syst 2002;**10**(5):571–88.
29. Dwork C, Roth A. The algorithmic foundations of differential privacy. Found Trends Theor Comput Sci 2013;**9**(3-4):211–407.
30. Sweeney L. Datafly: a system for providing anonymity in medical data. In: Lin TY, Qian S, eds. Database Security XI. Boston, MA: Springer; 1998:356–81.
31. Prasser F, Bild R, Eicher J, et al. Lightning: Utility-driven anonymization of high-dimensional data. Trans Data Priv 2016;**9**(2):161–85.
32. Lin J-L, Wei M-C. Genetic algorithm-based clustering approach for k-anonymization. Expert Syst Appl 2009;**36**(6):9784–92.
33. El Emam K, Dankar FK, Issa R, et al. A globally optimal k-anonymity method for the de-identification of health data. J Am Med Inform Assoc 2009;**16**(5):670–82.
34. Kohlmayer F, Prasser F, Eckert C, et al. Highly efficient optimal k-anonymity for biomedical datasets. In: 2012 25th IEEE International Symposium on Computer-Based Medical Systems (CBMS), Rome, Italy. 2012:doi:10.1109/CBMS.2012.6266366.
35. Kohlmayer F, Prasser F, Eckert C, et al. Flash: efficient, stable and optimal K-Anonymity. In: 2012 International Conference on Privacy, Security, Risk and Trust and 2012 International Confernece on Social Computing, Amsterdam, Netherlands. 2012:708–17.
36. Mitchell M. An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press; 1996.
37. Katoch S, Chauhan SS, Kumar V. A review on genetic algorithm: past, present, and future. Multimed Tools Appl 2021;**80**(5):8091–126.
38. Wan Z, Vorobeychik Y, Xia W, et al. Expanding access to large-scale genomic data while promoting privacy: A game theoretic approach. Am J Hum Genet 2017;**100**(2):316–22.
39. Prasser F, Gaupp A, Wan Z, et al.. An open source tool for game theoretic health data de-identification. AMIA Annu Symp Proc 2017;**2017**:1430–9.
40. Prasser F, Kohlmayer F, Kuhn K. The importance of context: risk-based de-identification of biomedical data. Methods Inf Med 2016;**55**(4):347–55.
41. Sankararaman S, Obozinski G, Jordan MI, et al. Genomic privacy and limits of individual detection in a pool. Nat Genet 2009;**41**(9):965–7.
42. Webdev EF. Eclipse Nebula - Supplemental Widgets for SWT. 2013. https://projects.eclipse.org/projects/technology.nebula. Accessed 7 July 2020.
43. Samarati P, Sweeney L. Generalizing data to provide anonymity when disclosing information (abstract). In: Proceedings of the seventeenth ACM SIGACT-SIGMOD-SIGART symposium on Principles of Database Systems - PODS '98, Seattle, WA. 1998:188.
44. Sharing Clinical Trial Data: Maximizing Benefits, Minimizing Risk (Appendix B Concepts and Methods for De-identifying Clinical Trial Data). Washington, D.C.: National Academies Press; 2015:18998.
45. Pitman J. Random discrete distributions invariant under size-biased permutation. Adv Appl Probab 1996;**28**(2):525–39.
46. Hoshino N. Applying Pitman's sampling formula to microdata disclosure risk assessment. J Off Stat 2001;**17**(4):499.
47. Iyengar VS. Transforming data to satisfy privacy constraints. In: Proceedings of the eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '02, Edmonton, Alberta, Canada; 2002:279.
48. UCI Machine Learning Repository: Adult Data Set. http://archive.ics.uci.edu/ml/datasets/adult. Accessed 7 July 2020.
49. American Time Use Survey Data Extract Builder. https://www.atusdata.org/atus/index.shtml. Accessed 7 July 2020.
50. IPUMS NHIS. https://nhis.ipums.org/nhis/. Accessed 7 July 2020.
51. US Census Bureau. American Community Survey (ACS). https://www.census.gov/programs-surveys/acs. Accessed 7 July 2020.
52. UCI Machine Learning Repository: default of credit card clients Data Set. https://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients. Accessed 7 July 2020.
53. Open psychology data: Raw data from online personality tests. https://openpsychometrics.org/_rawdata/. Accessed 7 July 2020.
54. Casas-Roma J, Herrera-Joancomartí J, Torra V. Comparing random-based and k-anonymity-based algorithms for graph anonymization. In: Torra V, Narukawa Y, López B , et al., eds. Modeling Decisions for Artificial Intelligence. Berlin, Heidelberg: Springer; 2012:197–209.
55. Solanas A, Martinez-Balleste A, Mateo-Sanz JM, et al. Multivariate microaggregation based genetic algorithms. In: 2006 3rd International IEEE Conference Intelligent Systems, London. 2006:65–70.
56. He Y, Naughton JF. Anonymization of set-valued data via top-down, local generalization. Proceedings VLDB Endowment 2009;**2**(1):934–45.
57. Fung BCM, Wang Ke, Yu PS. Top-down specialization for information and privacy preservation. In: 21st International Conference on Data Engineering (ICDE'05), Tokyo. 2005:205–16.

58. Fung BCM, Wang K, Chen R, et al. Privacy-preserving data publishing: A survey of recent developments. ACM Comput Surv 2010;**42**(4):doi:10.1145/1749603.1749605.
59. Nayak TK, Adeshiyan SA. On invariant post-randomization for statistical disclosure control: Invariant PRAM for disclosure control. Int Stat Rev 2016;**84**(1): 26–42.
60. Meurers T, Prasser F. Benchmark of ARX's Heuristic Algorithms. 2021. https://github.com/arx-deidentifier/genetic-benchmark. Accessed 18 July 2021.
61. Meurers T, Bild R, Do K-M, et al. Supporting data for "A scalable software solution for anonymizing high-dimensional biomedical data." GigaScience Database, 2021. http://dx.doi.org/10.5524/100929.