How (Not) to Protect Genomic Data Privacy in a Distributed Network: Using Trail Re-identification to Evaluate and Design Privacy Protection Systems

Bradley Malin Latanya Sweeney

April 2004
CMU-ISRI-04-115

Institute for Software Research International
Data Privacy Laboratory

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213-3890

Abstract

The increasing integration of patient-specific genomic data into clinical practice and research raises serious privacy concerns. Various systems have been proposed that protect privacy by removing or encrypting explicitly identifying information, such as name or social security number, into pseudonyms. Though these systems claim to protect identity from being disclosed, they lack formal proofs. In this paper, we study the erosion of privacy when genomic data, either pseudonymous or data believed to be anonymous, is released into a distributed healthcare environment. Several algorithms are introduced, collectively called RE-Identification of Data In Trails (REIDIT), which link genomic data to named individuals in publicly available records by leveraging unique features in patient-location visit patterns. Algorithmic proofs of re-identification are developed and we demonstrate, with experiments on real-world data, that susceptibility to re-identification is neither trivial nor the result of bizarre isolated occurrences. We propose that such techniques can be applied as system tests of privacy protection capabilities.

This research was supported by the Data Privacy Laboratory at Carnegie Mellon University.

Keywords: *Privacy, Anonymity, Re-identification, Genomics, DNA Databases*

1. INTRODUCTION

Modern medicine is currently in the midst of a genomics revolution that promises significant opportunities for healthcare advancement [1, 2]. At the same time, the increased incorporation of genomic data into medical records and the subsequent sharing of such data raise complex patient privacy issues. These issues have yet to be sufficiently addressed by the biomedical community. In general, the term privacy is semantically overloaded and now encompasses many distinct topics, which makes discussions of privacy difficult. This work addresses data anonymity, which provides provable assurances that data cannot be related to its subjects. This work does not address security and policy components of privacy that have been discussed in various health communities [3, 4, 5].

Recently, several identity protection solutions have been proposed to address the problem of anonymity. Many methods advocate the use of encrypted pseudonyms [6, 7] or the de-identification [8, 9] of explicit identifiers, such as name or social security number, initially associated with genomic data. However, these solutions lack proofs or guarantees of privacy afforded to the protected data. Contrary to popular belief, the protection of a patient's anonymity in genomic data is not as simple as removing, or replacing, explicit identifying attributes. Though genomic data may look anonymous, anonymity can only be guaranteed when inferences that can be garnered from genomic data itself are accounted for. While encryption and de-identification prevent the direct linking of genomic data to explicit identity, research presented in this paper contends that they provide a false appearance of anonymity.

Specifically, this work is concerned with genomic data scattered across a set of locations. In a distributed data sharing environment, patients visit and leave behind data at multiple data collecting locations, such as hospitals. Each location may sever genomic data from clinical data and, subsequently, release genomic data in order to enable such endeavors as basic research [10] and clinical trials [11, 12]. Therefore, it is in this environment, where we prove that the anonymity of the genomic data can be compromised. We develop and evaluate a general technique for re-identifying seemingly anonymous genomic data to the named individuals that the data was derived from.

Our work serves two main purposes. First, it raises awareness that anonymity protection methods must account for healthcare and medical inferences that exist in a data sharing environment. Second, this work provides the biomedical community with a formal computational model of a re-identification problem that pertains to genomic data. We believe that our models, as well as others [13, 14], can be applied as tests of the privacy protection capabilities of existing and developing privacy protection systems.

The remainder of this paper is organized as follows. In the following section, we present some deficiencies in current protection methods. We present a simple model of re-identification that this work builds upon. In Section 4, re-identification methods are formalized as a family of computational algorithms. In Section 5, we analyze how the algorithms perform with real world data. Finally, in Section 6 we discuss the limitations, possible extensions of our methods, and how this work can help researchers design more adequate anonymity protection techniques.

2. BACKGROUND

There are several reasons why current privacy protection methods fail to sufficiently protect the anonymity of genomic data. One reason is that they neglect to protect identifying inferences drawn from the genomic data itself. A second reason concerns the ability to relate genomic information to other publicly available information.

The ability to infer identifying features from genomic data is exemplified by our prior research into genotype-clinical phenotype relations. We developed a general model with the capability to learning patient-specific genomic data from publicly available longitudinal medical information [15]. The model relates a disease's symptoms to particular clinical states of the disease. Appropriate weighting of the symptoms is learned from observed diagnoses to subsequently identify the state of the disease presented in hospital visits. This approach is applicable to any simple genetic disorder with defined clinical

phenotypes. The efficacy of our model was demonstrated by inferring specific DNA mutations of clinically positive Huntington's disease patients. Our model utilized existing knowledge about the strong inverse correlation between the disease age of onset and the number of CAG repeat mutations in the HD gene.

In other previous research, we presented a specific scenario where genomic data, devoid of any identifiers, was uniquely re-identified, through an algorithm called REID (RE-Identification DNA), to the name and demographics of the patients that the data was collected from [16]. The REID algorithm exploits what we now refer to as the trail generated by occurrences of the data across independent hospitals. Releasing the genomic data alone, even devoid of pseudonyms, provides no guarantee of anonymity because the locations at which the genomic data appear can be compared to occurrences of patients at hospitals using hospital discharge data [17]. These trails of genomic data and trails of patient appearances in medical data can match uniquely.

The REID algorithm is limited in its scope, because genomic data re-identification can occur only if a strict set of assumptions hold. Therefore, in this paper we both generalize our original re-identification technique and introduce a family of re-identification trail re-identification methods that relax these assumptions for more general applicability.

3. DATA MODEL

The re-identification algorithms are best understood by structuring the data that is released by data holders. In this section we discuss the process by which data is so structured and the properties that appear in the data structures. We begin with an example of a data collecting and sharing example.

3.1. Scenario

Consider the following situation. John Smith is admitted to a local hospital, where he is diagnosed, via a DNA diagnostic test, with a DNA-influenced disease, such as cystic fibrosis. The hospital stores the clinical and DNA information in John's electronic medical record. For treatment, John visits several other hospitals, where his electronic medical record is collected and stored. For research purposes, the hospitals forward certain DNA databases, including John's DNA, onto a research group [1, 2]. The DNA records are tagged with the submitting institution and with pseudonyms for their submitted sequences [9]. By state law, the hospital sends a copy of the discharge record onto a state-controlled database. The discharge database is made publicly available in a de-identified format and can be re-identified [13, 18]. The availability and potential of re-identification remain even under the new medical privacy resolution known as the Health Insurance Portability and Accountability Act (HIPAA). We can track which hospitals John visited in the discharge data and we can track his DNA information in the research data. The sets of locations John visited we call a trail, and the uniqueness features of trails allow DNA trails in the research data to be matched to trails from their identified discharge database counterparts.

3.2. Basic Model

The basic model elements are derived from relational database theory. The term *data* refers to information held by a data-collecting location, such as a hospital. The data is organized as a table $\tau(A_1, A_2,\ldots,A_p)$, with attributes $A = \{A_1, A_2,\ldots,A_p\}$. Each row is a *p-tuple* consisting of patient information $t[a_1,\ldots,a_p]$, and represents the sequence of values, $a_1 \in A_1,\ldots,a_p \in A_p$. The size of the table is simply the number of tuples and is represented $|\tau|$. In our model, each data-collecting location releases a 2-table[1]

[1] Actually, this is a specific case of re-identification from an *n*-table vertical partitioning, where data is released by a single location as *n* different tables. This problem is the same as the 2-table problem, except it requires iterating the entire re-identification process *n*-1 times. For simplicity sake, we present only 2-table re-identification problem.

vertical partitioning of its data table. The first table, τ^+, is called the *identified subtable* and contains explicitly identified data (e.g. name, address, social security number, etc…) with attributes A^+, where $A^+ \subseteq A$. The second table, τ^-, is called the *DNA subtable* and consists of DNA information only, with attributes $A^- \subseteq A$.

τ							
τ^+						τ^-	
Name	**Birthdate**	**Sex**	**Zip**	**Diagnosis**	**Treatment**	**Pseudonym**	**DNA**
John Smith	2/18/45	M	15234	3330	132	SA9212OK19	cttg…a
Mary Doe	4/9/75	F	15097	33520	653	AS09D8LK1J	atcg…t
Bob Little	2/26/49	M	15212	27700	742	D8A79AD133	acag…t
Kate Erwin	11/3/54	F	15054	3563	123	ASSD834MS1	accg…a

Fig. 1. **Vertical data partitioning into an identified table (τ^+) of patient demographics and a DNA table (τ^-) containing de-identified sequences.**

As an example, consider the database records in Fig. 1, where generic clinical data is stored in τ^+ and electronic DNA sequences are stored in τ^-. Notice that at the location housing the database, the relationship between DNA and identities is explicitly known, while in the partitioned release the order of the tuples may be changed.

Before continuing, several assumptions about the environment should be noted. First, it is assumed that each data collecting location releases data that was collected by itself and from no external source. Therefore, it is not possible for hospital *H* to release the DNA sequences of patient *X* if patient *X* never visited hospital *H*. Second, tuples released in the de-identified and identified tables are unique for each patient. Though a patient may visit a hospital on multiple occasions, the information released by the hospital corresponds to a patient, but not to the frequency of the patient's visits to a hospital.

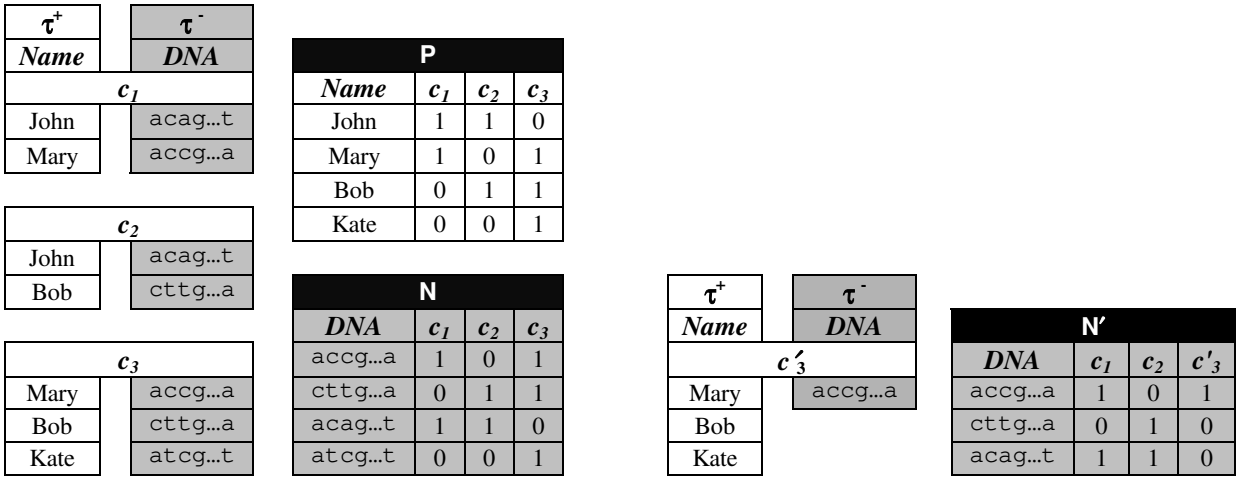
3.3. Data Structures

The static nature of patient demographics and genomic information allows for data to be followed across releases from different locations. We make the tracking of data explicit by constructing two matrices. The first matrix is called the *DNA track* **N**, and consists of information pertaining to shared DNA data. The dimensions of this matrix are $|\cup_{c \in C} \tau_c^-| \times (|A^-| + |C|)$ and each row in this matrix corresponds to a unique DNA sample released by the set of locations. The cells of the first $|A^-|$ columns of the matrix represent the DNA information collected from τ_c^-. The latter $|C|$ cells are Boolean representations of the DNA data at each location. Values associated with the locations are 1 if the DNA sample was released from the location and 0 otherwise. The second matrix is called the *identified track* **P** and is similar to the first matrix, except it maintains a representation of the identified data in the first $|A^+|$ cells. For a more concrete example, the data releases of three locations and the corresponding tracks **P** and **N** are provided in Fig. 2.

When every location releases tables, such that the only tuples present in τ^- have corresponding tuples in τ^+, and vice versa, we say that the tracks are *unreserved*. The tracks **P** and **N** in Fig. 2 are unreserved. However, both data releasers and patients are autonomous entities, and either can choose to withhold certain information. Thus, releases that are unreserved are not always practical and, at times, can be impossible. Consequently, we say that track **N** is *reserved* to track **P** if for every location *c*, every tuple $x \in \tau_c^-$ there exists a tuple $y \in \tau_c^+$, such that both *x* and *y* are derived from the same tuple in τ. Similarly, **P** can be reserved to track **N**. By substituting c'_3 for c'_3, in Fig. 2, the DNA track **N′** is reserved to the identified track **P**.

The vector of binary values associated with the latter $|C|$ attributes, we refer to as a *trail*. We denote a trail for data *d* in an arbitrary track **T** as *trail*(**T**,*d*). When a trail resides in an unreserved track, it is called

a *complete* trail because the binary values unambiguously convey the presence or absence of a patient at a location. When a trail exists in a reserved track (e.g. **N′** of Fig. 2) it is called an *incomplete* trail, since the value of 0 is ambiguous.



τ^+	τ^-
Name	*DNA*
c_1	
John	acag…t
Mary	accg…a

c_2	
John	acag…t
Bob	cttg…a

c_3	
Mary	accg…a
Bob	cttg…a
Kate	atcg…t

P			
Name	c_1	c_2	c_3
John	1	1	0
Mary	1	0	1
Bob	0	1	1
Kate	0	0	1

N			
DNA	c_1	c_2	c_3
accg…a	1	0	1
cttg…a	0	1	1
acag…t	1	1	0
atcg…t	0	0	1

τ^+	τ^-
Name	*DNA*
c'_3	
Mary	accg…a
Bob	
Kate	

N′			
DNA	c_1	c_2	c'_3
accg…a	1	0	1
cttg…a	0	1	0
acag…t	1	1	0

***Fig. 2. Left*) Identified (P) and DNA (N) tracks created from unreserved releases. Both P and N are unreserved tracks. *Right*) Resulting DNA track created from a reserved release. N is now reserved to P.**

Through the ambiguity present in the 0 value, there is a simple relationship between a patient's incomplete trail and complete trail. We say that a trail x is a *subtrail* of trail y ($x \leq y$) if for every value of 1 in x, there is a value of 1 in y. Similarly, y is the *supertrail* of x. The ambiguity prevents a direct mapping of an incomplete trail in one track to its complete trail in the other track. This is because, given an incomplete trail made up of n locations with m 0's, there are 2^m potential complete trails that the incomplete trail could be mapped to. For example, using tracks **P** and **N′** from Fig. 2, *cttg...a*[0,1,0] and *acag...t*[1,1,0] are subtrails of *John*[1,1,0]. Similarly, *John*[1,1,0] and *Bob*[0,1,1] are supertrails of *cttg...a*[0,1,0].

We have now described the data sharing environment, the data structures, and their formal properties. In the following section, we provide a set of algorithms that utilize these data structures and properties for re-identification purposes.

4. RE-IDENTIFICATION ALGORITHMS

Given the tracks constructed above, the trail re-identification problem is how to properly and uniquely link identified data to DNA data through common features in their trails. In this section will provide algorithms for doing exactly this. The two algorithms presented in this section are collectively termed Re-identification of Data in Trails (REIDIT), since each exploits a different aspect of the relationships between trails.

4.1. REIDIT-Complete

The first re-identification algorithm is called REIDIT-Complete, or REIDIT-C. REIDIT-C performs exact matching on the trails in tracks **N** and **P**. It assumes that both **N** and **P** are unreserved, and therefore, is only applicable with complete trails. The pseudocode of REIDIT-C is provided in Fig. 3. For every tuple in $n \in \mathbf{N}$, REIDIT-C determines if there exists one and only one tuple $p \in \mathbf{P}$ such that *trail*(**N**,n) equals *trail*(**P**,p). When there is an exact and unique match, then the genomic data of *trail*(**N**,n)

is re-identified to explicitly identifying information in **P**. If *trail*(**N**,*n*) is equivalent to both *trail*(**P**,*p*) and *trail*(**P**,*p′*), where $p \neq p'$, then there is an ambiguity and no re-identification can occur.

REIDIT-C Algorithm

Input: DNA and Identified Tracks **N** and **P** for the same data-collecting locations.
Output: Set of trail re-identifications *Reidentified*
Assumes: N and **P** are unreserved
Steps:

```
let Reidentified be an empty set
for each tuple n in N
    if there exists only one tuple p in P, such that trail(P,p) ≡ trail(N,n)
        Reidentified = Reidentified ∪ [p, n]
return Reidentified
```

Fig. 3. **Pseudocode for REIDIT-C.**

REIDIT-C can generate the four possible results for two arbitrary trails *trail*(**N**,*n*) and *trail*(**P**,*p*), as shown in Table 1: 1) correct match, 2) correct non-match, 3) false non-match, and 4) false match. The first three can occur, while the last is impossible. The reasoning is as follows. One of the main assumptions of the unreserved-release model is that both trails in **P** and **N** are complete. Therefore, a correct match can only be made when *trail*(**N**,*n*) ≡ *trail*(**P**,*p*). When there is only one equivalent trail in **N** for *trail*(**P**,*p*), as well as only one equivalent trail in **P** for *trail*(**N**,*n*), then this must be a correct match. In the event that, there are multiple equivalent trails, then for *trail*(**N**,*n*) there will be a set of equivalent trails in **P**, one of which must be a correct match. Since the correct trail is indistinguishable from the incorrect trails, no match will be made. To prevent a false match from being assigned, a false non-match will occur. When *trail*(**N**,*n*) ≠ *trail*(**P**,*p*), then the two trails can not refer to the same entity, and thus a correct non-match will be made. Therefore, for each trail in **P** there must exist a minimum of one equivalent trail in **N**. If there is only one equivalent trail in **N**, then both the trail in **P** and the trail **N** must correspond to the same individual.

	Re-identification	**No Re-identification**
***trail*(N, *n*) = *trail*(P, *p*)**	True match	False non-match
***trail*(N, *n*) ≠ *trail*(P, *p*)**	False match	True non-match

Table 1. Classification of re-identifications made by REIDIT-C. Light-shaded cells are possible outcomes and the darkened cell is an impossible outcome.

First, recall the underlying assumption of the unreserved-release model: tuples of both tracks **N** and **P** consist only of complete trails. Thus, at website *w*, a visit from an entity must be recorded in both T_w^- and T_w^+. Since this holds true for every website, for each *trail*(**N**,*n*), there must exist at minimum one equivalent *trail*(**P**,*p*). If there exists more than one equivalent trail in **P** for *trail*(**N**,*n*), then multiple trails will be recognized and the singleton requirement will not be satisfied. No re-identification will be recorded.

The computational complexity of REIDIT-C, as presented in Fig. 3, is quadratic in the size of the DNA table, $O(|\mathbf{N}|^2)$. We can count the number of steps as follows. First, the outer loop iterates over all of the tuples in **N**, which is |**N**| iterations. Second, for each iteration in **N**, the algorithm iterates a maximum of |**P**| times. This provides $O(|\mathbf{N}|\bullet|\mathbf{P}|)$, which equals $O(|\mathbf{N}|^2)$ because |**N**|=|**P**|. However, the quadratic bound is an artifact of the way in which the pseudocode is written. Another version based on sorting could be written, such that both set of trails are sorted and then compared. Though more complex in the data structure, the new version would produce a complexity bound of $O(|\mathbf{N}|\log|\mathbf{N}|)$.

4.2. REIDIT-Incomplete

The second re-identification algorithm is named REIDIT-Incomplete, or REIDIT-I. It is applicable when one track is reserved to the other. Fig. 4 provides pseudocode and commentary for the algorithm. The algorithm works as follows. For each trail in the track containing incomplete trails, the set of its supertrails from the other track are determined. If there is only one supertrail, then a correct re-identification has occurred. The re-identified trails from **N** and from **P** are then removed. The removal of the re-identified trails is a crucial step. Since the complete trail can have multiple subtrails, failure to remove the trail from consideration can prevent additional trails from being re-identified. This process continues until no more re-identifications can be made because one of two conditions is satisfied: either (1) the track with incomplete trails has no more trails to process; or, (2) there are no re-identifications made in the current iteration.

REIDIT-I can generate the four possible results for two arbitrary trails *trail*(**N**,*n*) and *trail*(**P**,*p*), as shown in Table 2: 1) correct match, 2) correct non-match, 3) false non-match, and 4) false match. The first three can occur, while the last is impossible. The reasoning is as follows. One of the main assumptions of the reserved-release model is that trails in **N** are incomplete, which means that only the 1's of the trails can be trusted. Regardless, it must be true that for an arbitrary trail in **N**, there must exist a non-null set of supertrails in **P**. If the set of supertrails is of size one, then this must be a correct match. In the event that there are multiple subtrails no re-identification will be made in the current iteration. Yet, in the current, and subsequent iterations, the set size may be reduced. The minimum set size is equal to 1, since there must exist at least one supertrail for the trail in question. When the set size does equal 1, then a correct re-identification will be made. If the set size can not be reduced to 1, then a false non-match will occur. In the case that *trail*(**N**,*n*) is not a subtrail of *trail*(**P**,*p*), it is not possible for a re-identification to be made. Thus, for any two trails *trail*(**N**,*n*) and *trail*(**P**,*p*), where *trail*(**N**,*n*) is not a subtrail of *trail*(**P**,*p*), only true non-matches will be recorded.

Algorithm: REIDIT-I-Fast (X, Y)

Input: DNA and Identified Tracks **N** and **P** for the same data-collecting locations. **X** is the reserved table of **N** and **P**, and **Y** is the remaining table.

Output: Set of trail re-identifications *Reidentified*

Assumes: 1) **X** has incomplete trails and **Y** has complete trails. 2) **X** is the reserved track of **Y**

Steps

```
let Z be a |X| × |Y| matrix, such that Z[x,y] = 1 if trail(X,x) ≤ trail(Y,y) and 0 otherwise
let S be a |X| × 1 column vector, such that S[x] is the sum of the x^th row of Z
let Reidentified be an empty set
let FoundOne = False
do
    FoundOne = False
    for x=1 to |X|                                          // find an incomplete trails that
        if S[x] ≡ 1                                         // has only one supertrail
            FoundOne = True
            for y=1 to |Y|                                  // if found, find the supertrail and
                if Z[x,y] ≡ 1                               // add the [supertrail,subtrail] pair
                    Reidentified = Reidentified ∪ [y, x]    // to the re-identified set
                    for z=1 to |X|                          // remove the re-identified
                        if Z[z,y] ≡ 1                       // supertrail from further
                            Z[z,y] = 0                      // consideration
                            S[z] = S[z] – 1
while FoundOne ≡ True
return Reidentified
```

Fig. 4. **Pseudocode for REIDIT-I-Fast, a version of REIDIT-I with an efficient data structure.**

For a complexity analysis of REIDIT-I, let **N** be reserved to **P**. From a computational standpoint, the REIDIT-I algorithm is the basic structure of REIDIT-C with an additional outer loop. Thus, by a simple extension to the complexity proof of REIDIT-C, we can potentially iterate $|\mathbf{N}|$ times, and it appears that the complexity of REIDIT-I is $O(|\mathbf{N}|^2 \bullet |\mathbf{P}|)$. However, we can abstract information in such a way that the complexity can be reduced to $O(|\mathbf{N}| \bullet |\mathbf{P}|)$. This method we call REIDIT-I-Fast and which is depicted in Fig. 4.

	Re-identification	**No Re-identification**
***trail*(N, *n*) ≤ *trail*(P, *p*)**	True match	False non-match
Not (*trail*(N, *n*) ≤ *trail*(P, *p*)	False match	True non-match

Table 2. Classification of re-identifications made by REIDIT-I. Light-shaded cells are possible outcomes and the darkened cell is an impossible outcome.

Consider an adjacency matrix Z of size $|\mathbf{N}| \times |\mathbf{P}|$, where each cell $Z[n,p]$ has a value of 1 if $trail(\mathbf{N},n) \leq trail(\mathbf{P},p)$. In addition, let S be a column vector of size $|\mathbf{N}|$ where each cell is the rowsum of Z. Construction of the matrix and vector occurs in approximately $O(|\mathbf{N}| \bullet |\mathbf{P}|)$ steps. In the do-while loop, the worst-case scenario occurs when each iteration yields one re-identification, thus taking $|\mathbf{N}|$ iterations. Within the loop, a sequential scan of the S vector takes place in $|\mathbf{N}|$ steps. If a unique re-identification is found, realized when $S[x]$ is 1, then a scan of one row of the Z matrix occurs using the inner for loop; this takes $|\mathbf{P}|$ steps. When cell $Z[x,y]$ with value 1 is found, the found column in Z and the S vector are updated with a scan taking $|\mathbf{N}|$ steps. Since, in worst case there is only one re-identification per do-while iteration, this process only occurs once per iteration. Thus, the total number of steps for the while loop and its internal processes is approximately $|\mathbf{N}| \bullet (2 \bullet |\mathbf{N}| + |\mathbf{P}|)$, which is approximately $O(|\mathbf{N}|^2 + |\mathbf{P}| \bullet |\mathbf{N}|)$. Therefore, the order of complexity will be $O(setup) + O(scanning)$ and since $|\mathbf{P}| \geq |\mathbf{N}|$, complexity is $O(|\mathbf{N}| \bullet |\mathbf{P}|)$.

4.3. Upper Bounds

Since a trail is vector of Boolean values, the set of trails can be discussed in terms of binary strings. For both REIDIT-C and REIDIT-I, the maximum number of trail re-identifications is dependent on the number of permutations of a binary string. Let C be the set of data releasing location and **P** be the identified track. The maximum number of trail re-identifications is bounded by the minimum of $|\mathbf{P}|$ and $2^{|C|}$-1. When $|\mathbf{P}| \leq 2^{|C|}$-1, then the maximum number of trail re-identifications is bounded by $|\mathbf{P}|$, which is the number of distinct patients in the considered population. This implicates that all trails may be re-identified. When $|\mathbf{P}| > 2^{|C|}$-1, the maximum number of trail re-identifications is bounded by the number of different binary location visit patterns that can be generated from $|C|$ locations.

5. EXPERIMENTS

Though in theory the re-identification limits of REIDIT-C and -I scale exponentially, this does not typically occur in the real world. A main contributing factor is that people do not visit locations in a random manner. On the contrary, many healthcare factors influence where an individual leaves data behind. For example, many hospitals have referral programs, such that there is nontrivial correlation between the visits of several hospital visits. Moreover, people tend to visit hospitals that are within close proximity to their residence. A hospital that is situated in the middle of a city will see more patients than a hospital in a rural setting. In addition, certain hospitals offer specialized care or treatment for particular diseases. Given these, and additional idiosyncrasies of the real world, REIDIT must be evaluated with real health data.

5.1. *Description of Real World Data*

The dataset used for evaluation consists of publicly available hospital discharge data from the State of Illinois, for the years 1990 through 1997. There are approximately 1.3 million hospital discharges per year and collection has compliance with greater than 99% of discharges occurring in hospitals in the state [13]. Typical discharge data is made up of demographic and clinical information. The demographic data includes date of birth, gender, zip code of residence, and hospital visited, while clinical information per patient visit includes a set of one to nine International Classification of Disease, Version 9 (ICD-9) codes and procedure codes.

From the discharge databases, longitudinal medical profiles for patients diagnosed with genetic disorders were constructed as follows. First, the set of patients that were diagnosed with a single gene disease was determined. A patient was represented by a distinct combination of the demographic values {*date of birth*, *gender*, *five digit zip code*}. Next, the databases were requeried with the previous demographic data to append additional clinical information from other hospital visits. Profiles were then probabilistically merged based on census demographics for {*age*, *gender*, *zip code*}, such that profiles likely to relate to the same person were combined. The uniqueness of patient identities making up profiles was 98-100% based on census data as reported previously [20]. Demographic data is considered to be identifying information, since each unique patient can be re-identified by simple linkage on demographics to publicly available identified data, such as voter registration lists [13]. In prior research we discovered that standard ICD-9 codes leak DNA-related data [20], such as genetic disorders and gender. We utilize both of these features in our analysis.

5.2. *Re-identifiability with REIDIT-C*

Eight populations afflicted with single gene disorders are analyzed. These populations are cystic fibrosis (CF), Friedrich's Ataxia (FA), hereditary hemorrhagic teleganictasia (HT), Huntington's disease (HD), phenylketonuria (PK), Refsum's disease (RD), sickle cell anemia (SC), and tuberous sclerosis (TS).

To evaluate re-identification with REIDIT-C, we make the following assumption about patient data. It is assumed that if a discharge profile specifies that a patient made a visit to a particular hospital, then both clinical and DNA data are released by the hospital about the patient. REIDIT-C was used with the set of profiles for each of the eight populations and gender-specific subpopulations. As specified in the previous section, all re-identifications returned by REIDIT-C are a true match. The results are presented in Table 3.

Since, the number of patients, for each population, is less than two to the number of total hospitals visited, the maximum number of re-identifications in theory is the number of patients. However, the observed number of re-identifications only achieves this maximum for the RD population, where there is only one patient with the disease at each of the hospitals considered. For the remaining populations, it appears that healthcare factors have a profound effect on the uniqueness of trails. A quick inspection reveals that the re-identifiability of these populations is related to the average number of patients visiting a hospital. This effect is graphically depicted in Fig. 5. It is apparent that as the number of people per hospital increases, the more difficult it is for re-identifications to occur. This phenomenon is due, in part, to the fact that an increase in population size, over a fixed set of locations, increases the probability that multiple patients will have the same trail. The average number of patients per hospital is a gross measure of re-identification. There are additional features about the environment that affect the re-identifiability of a population, which we expect to explore in future studies.

The belief that each location in a health environment will collect and release genomic data may be unrealistic given the current state of the health care market. Though such an environment may exist in the future, we must consider a more fine-grained perspective by analyzing how particular locations and sets of locations can affect the re-identifiability of patients in a population. It is more realistic that only a fraction of hospitals will be releasing genomic data about patients. As exemplified in Figure 5, the

number of patients per location affects re-identifiability. Yet, this does not indicate which locations have an effect.

Disease	Gender	Number of Patients	Number of Hospitals	Average Number of Patients Per Hospital	% Re-identified
CF		1149	174	11.92	32.90%
	Female	557	142	7.28	43.09%
	Male	592	150	6.94	39.36%
FA		129	105	2.08	68.99%
	Female	60	68	1.47	80.00%
	Male	69	72	1.65	78.26%
HD		419	172	4.37	50.00%
	Female	236	149	2.76	79.14%
	Male	183	127	2.70	50.63%
HT		429	159	4.83	52.21%
	Female	244	140	3.06	64.34%
	Male	185	114	2.98	63.24%
PK		77	57	2.15	75.32%
	Female	52	48	1.85	80.77%
	Male	25	25	1.36	80.00%
RS		4	8	1	100.00%
	Female	2	4	1	100.00%
	Male	2	4	1	100.00%
SC		7730	207	88.89	37.34%
	Female	4175	189	55.87	43.76%
	Male	3555	191	41.01	36.51%
TS		220	119	3.82	51.60%
	Female	97	88	2.60	78.35%
	Male	123	87	2.60	61.79%

Table 3. Susceptibility of Genetic Disease Populations to REIDIT-C Re-identification.

To answer this question, we study the effect of location popularity on re-identifiability of a population. We investigate the case where a certain set of locations are releasing data. More specifically, as can be seen in Figure 6, we consider an environment where an increasing number of hospitals participate in representative data sharing. We compare the re-identifiability for CF, where the number of patients per location is relatively large (~11.92), to PK, where the average is closer to a single individual per location (~2.15). Each hospital is ranked by the number of distinct patients visiting the location. A total rank ordering of the locations was achieved by randomly ordering locations with the same number of patients.

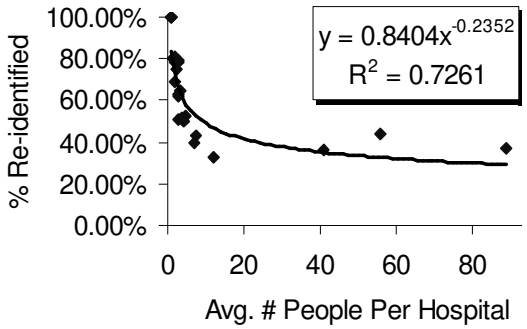


Fig. 5. **REIDIT-C re-identification of populations as a function of the average number of people per location.**

Given a set of locations from highest rank (*i.e.* most popular) down to a particular rank *x*, we measured the re-identifiability of the trails that were discovered (i.e. non-null trails over the set of locations ranked 1 to x). For both CF and PK, the rate of trail discovery is logarithmic as can be seen in Fig. 6. The r^2 correlation coefficients for fit curves were 0.92 and 0.97, respectively. However, while the rate of trail re-identification for CF is logarithmic ($r^2 = 0.92$), the rate for PK is linear ($r^2 = 0.98$). It appears that this is an artifact of the slope in the logarithmic discovery rate. The slope of trail discovery for CF is much greater than for PK. This implies that most individuals visited the more popular locations for CF, while for PK patients are more dispersed in hospitals.

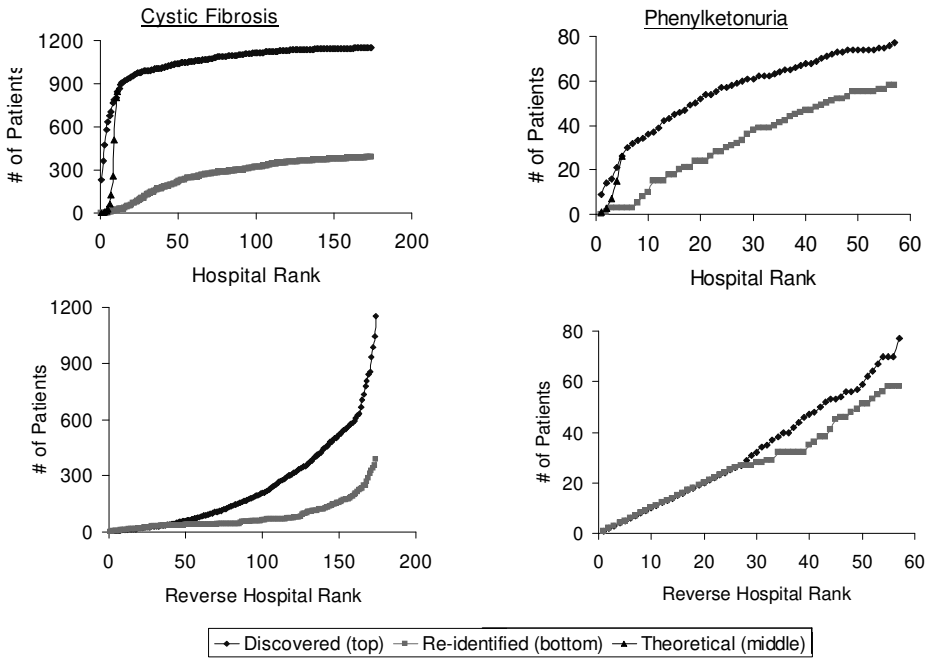


Fig. 6. **REIDIT-C re-identification as a function of hospital rank by visit popularity; *first row*) in order, *second row*) reverse order.**

One would expect that incorporation of less popular locations would make re-identification easier and that more popular locations would make re-identification more difficult. To evaluate this claim, we added locations in reverse rank, and measured the re-identifiability of the non-null trails constructed from the contributing locations. We find that for the first quarter of reverse rank websites, almost all patients in the population are re-identified. This is due to the fact that for most of these hospitals, the number of patient trails found and the number of re-identifications increases approximately linearly with slope equal to 1. This means that at these locations, usually only one patient existed at the hospital with the disorder. Thus, the first part of our hypothesis is true. After the first quarter locations, the re-identification rate for PH remains linear, with a slightly lower rate than the rate of trail discovery. However, the trail discovery rate for CF becomes exponential, and subsequently, after a delay, so too does the CF trail re-identification rate. This is due to the fact that as the number of people per location increases, the ability to distinguish a larger number of trails increases as well.

5.3. Re-identifiability with REIDIT-I

For analysis of REIDIT-I, we continue with the CF population profiles from above. The CF complete trails were used to generate incomplete trails for analysis of the REIDIT-I algorithm. To do so, we utilize a simple model of how locations create reserved releases. Each location withholds identifying information on a patient with the same probability *x*. Thus, the track of complete information consists of identified

clinical data trails and the track of incomplete information consists of genomic data trails. We varied the probability of information being withheld and attempted re-identification with REIDIT-I. As specified in the previous section, all re-identifications returned by REIDIT-I are a true match. Graphs of the results for x equal to 0, 0.1, 0.5, and 0.9 are shown in Fig. 7. Each point of a graph depicts the average result for 10 experiments of random information withholding.

As the probability of withholding information increases, the probability that an individual will not show up at all (i.e. no trail generated) in the population of incomplete trails. Thus, in the graphs we show three lines. The topmost line represents the number of non-null identified clinical data trails for a given set of hospitals. The middle line represents the number of non-null genomic data trails. And the lowest line represents the number of genomic data trails that were re-identified. As expected, we find that as the amount of information withheld increases, the number of releasing locations necessary to perform re-identification increases as well. This is due to the fact that as additional information is withheld, the incomplete trail becomes less complex and informative. However, even though trails become less complex, there remains a significant disposition toward re-identification. This is observable even after 50% of a trail is obscured. We find that there is an inverse relationship between the slope of re-identification (as a function of website rank) and the amount of information withheld.

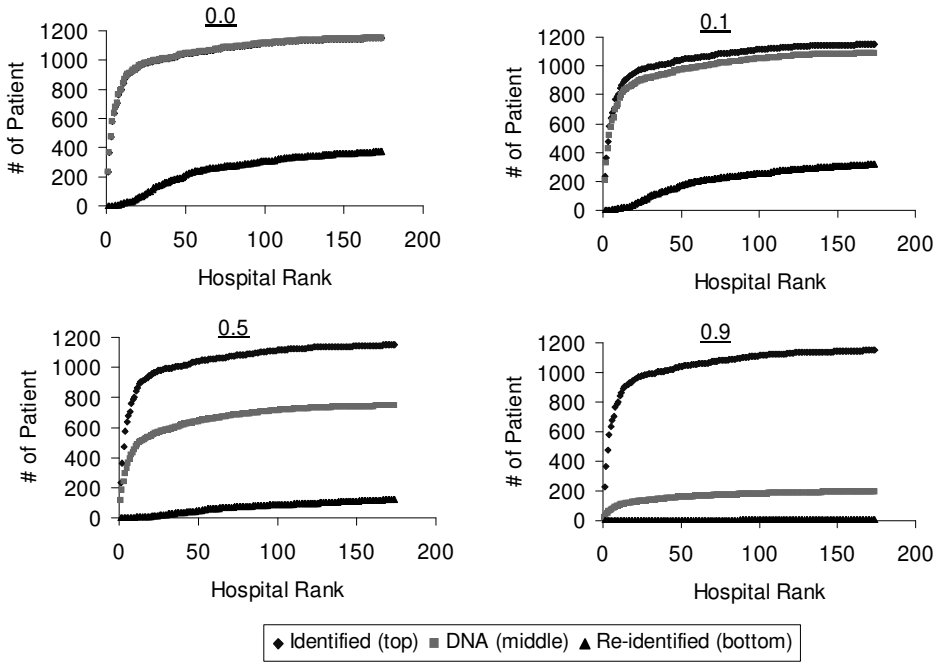


Fig. 7. **Re-identification of CF incomplete trails with REIDIT-I as an increasing amount of identifying information is withheld from the release. From left to right: 0.0, 0.1, 0.5, and 0.9 probability of withholding.**

6. DISCUSSION

Appearances can be deceiving. This concept has been uttered by countless people in many different eras, but it characterizes genomic data as well. Simply because genomic data is de-identified or pseudonymized does not mean that anonymity can be assumed. It is necessary that features about the data, as well as the environment in which the data is shared, are taken into account before data can be declared as anonymous. The REIDIT algorithms described in this paper are a prime example of how and why techniques that function in one environment, such as the use of encryption to protect security, can not be blindly relied upon to protect anonymity.

6.1. Privacy Protection Systems Testing

Various privacy protection schemas have been published and deployed for genomic data. These methods utilize protections such as encrypted pseudonyms provided by trusted third parties [6, 9] or de-identification of explicit identifiers [8, 21]. Each claims that it protects the privacy of the data subjects. While advocates of such techniques recognize that there exist re-identification threats from inferences about data itself [9], they deem such threats as minimal and unjustifiable as an impediment to research. Our experimental results demonstrate otherwise; the re-identification risk of de-identified data is non-trivial.

Though privacy protection schemas do not explicitly model protection against trail re-identification, not all schemas are susceptible to the attack. Here, we analyze two protection models, one that is susceptible to trail re-identification and one that is not. One susceptible model has been proposed by de Moor, Claerhout, and de Meyer [9]. In this model, a set of data holders, such as a set of hospitals, transfer data to a central repository maintained by a trusted third party. Both parties encrypt the identifying information associated with the DNA data. For a set of locations *A*, *B*, …, *Z*, the trusted third party maintains a set of datasets $\boldsymbol{A}^{-}\{g(f_A(Identity_A)), DNA\}$, $\boldsymbol{B}^{-}\{g(f_B(Identity_B)), DNA\}$, …, $\boldsymbol{Z}^{-}\{g(f_Z(Identity_Z)), DNA\}$, where g is the encryption function of the trusted party, f_i is an encryption function for location i, and $Identity_i$ is the set of identifying attributes used by location i for the encrypted pseudonym. When a new researcher requests sTTP for data, sTTP supplies the appropriate set of doubly-encrypted lists. This method protects direct access to the identity of the individual, but completely neglects the DNA data. A DNA track can easily be constructed from the released information. When identified clinical information is subsequently shared, an identified track can also be constructed. With a DNA track and an Identified track structured from multiple locations, trail re-identification can be conducted. It should be noted that masking the identity of the data location does not necessarily prevent trail re-identification. For example, in Fig. 8, an unreserved release is made, but the DNA datasets do not have locations explicitly listed. The ordering of bits in trails for the identified and DNA tracks are not necessarily the same. Regardless, a correct match on trails can be made by using the number of locations visited.

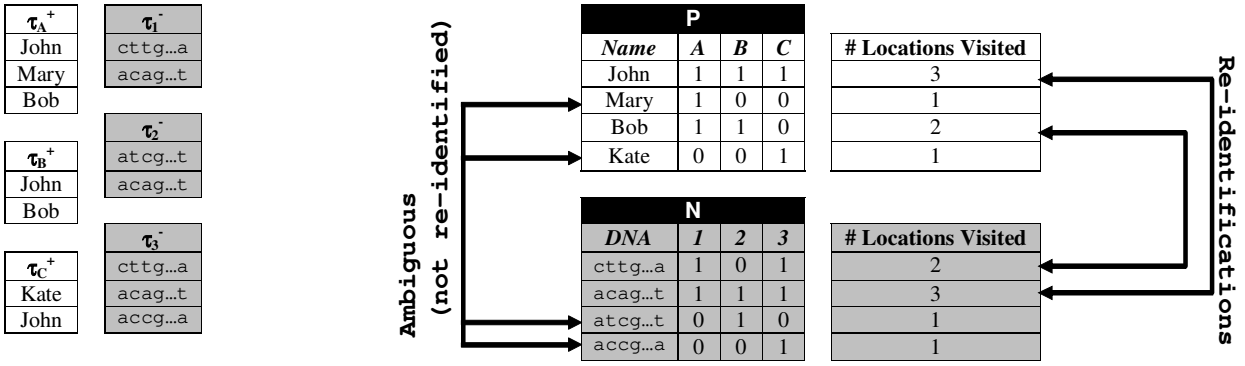


Fig. 8. ***Left*****) Unreserved release where locations are not explicitly identified.** ***Right*****) Resulting identified (P) and DNA (N) tracks. Re-identifications are made through uniqueness in the number of locations visited.**

In contrast, consider a privacy protection model proposed by deCODE Genetics, Inc. [6] of Iceland. deCODE researchers determine, with the assistance of physicians that attend to the general population, a set of individuals of research interest. The set of participating patients donate a blood sample at a facility run by the Data Protection Commission (DPC) of Iceland. The patients' Social Security Number is encrypted (using strong encryption) into a pseudonym, and is forwarded with the sample onto deCODE. In this system, an individual's clinical information is distributed and annotated with location information from multiple locations, thus an identified track can be constructed. However, an individual's DNA is

collected and annotated with one location only. Even if there are multiple locations run by the DPC for data collection, each individual's DNA trail will have a solitary location. Thus, the only susceptibility this system reveals to trail re-identification is when a single individual visits only one DPC location.

To be susceptible to trail re-identification does not imply that a protection model is impregnable to re-identification. In the following section, we briefly discuss additional susceptibility tests that can be employed.

6.2. Alternate Re-identification Models

Obviously, the REIDIT algorithms do not re-identify all genomic data samples. But does this guarantee that the unidentified data is anonymous? While it would be nice to unequivocally proclaim yes, this would be extremely naïve. While the REIDIT algorithms provide a single model of how re-identification can occur in a distributed environment, however, trail re-identification is not the only manner by which genomic data can be re-identified. An earlier re-identification model we introduced utilizes features about the genomic data [15] and simple relationships that may exist between DNA and clinical information (i.e. this sample contains a mutation for cystic fibrosis). Currently, one of the main focuses of research in personalized medicine is the study of how variation in an individual's genome affects their clinical phenotype [1, 22]. Though useful for research and clinical healthcare purposes, these same relationships also pose challenges to personal privacy.

For example, in previous work we demonstrated that specific DNA sequences of an individual's genomic data could be inferred from publicly available longitudinal clinical information [15]. In the study, we utilized a subset of the patient profiles of the Huntington's disease patients described previously. The identities of Huntington's Disease patients were determinable. The relation of each person's genomic information to their publicly available clinical information proceeds as follows. Through an intelligent model we were able to determine a small bound for the age of onset of the disease for the patients. Since there is a strong correlation between the age of onset and the size of the CAG repeat mutation that causes Huntington's disease, we were able to correctly infer the CAG repeat for 19 of 22 patients in the study. It is feasible that the models we utilized, or other models [13, 14], could be employed to infer the genomic information of individuals diagnosed with other genetic diseases and thereby re-identify the genomic information.

6.3. Limitations and Future Research

Though the REIDIT algorithms provide correct re-identifications, they are limited by their assumptions. First, in the reserved release model assumes that only one of the data types is reserved. If one location withholds genomic data, then all locations withhold genomic data. Yet, if one location withholds genomic data, and a different location withholds identified data, then both constructed tracks will consist of incomplete trails. In this scenario, trails from either track can have their 0's can be truthfully flipped to 1's in. The deterministic REIDIT-I algorithm can not handle such a scenario. Use of the REIDIT-I algorithm can result in an increased number of false negatives or missed re-identifications. Even worse, REIDIT-I may cause false re-identifications, which under the current error-free model is impossible to achieve.

Second, the model assumes that the released data is error-free. However, this may not be the case. In certain cases, typographical errors or false recordings of information in a database may occur. In this situation, not only can a 0 in a trail be flipped to a 1, but a 1 in a trail can correctly be flipped to a 0. Again, the REIDIT-I algorithm can miss and cause false re-identifications.

In light of these deficiencies, we are developing more robust trail re-identification algorithms. One possible direction is the development of trail re-identification methods based on record linkage models. Record linkage has been used in the biomedical community to link records from one database to records from another database. In [24], a deterministic record linkage model is proposed, where features selection of the best linkage attributes are determined. More complex record linkage model incorporate

probabilistic models to account for typographical error [25, 26]. For instance, "John H. Smith" in Database 1 and "Jon H. Smitth" in Database 2 may both be the same individual, but neither John and Jon, nor Smith and Smitth, are not equivalent. Variations on these probabilistic methods may be useful for designing new trail matching models. For example, consider a simple reserved release: an identified track with two trails, s_1[1,0,1] and s_2[0,1,1], and a DNA track with two trails, t_1[0,0,1] and t_2[1,1,1]. If each location has an equal amount of error in their released data, then no matches of identified to DNA trails can be made; both s_1 and s_2 differ from t_1 and t_2 by 2 bits. However, when the first location is known to have a high rate of data error and the remaining locations have little or no error, then it is more probable that s_1 and t_1 correspond to the same entity, and similarly for s_2 and t_2. Granted, the ability to make such a decision must be made in the context of the set of all trails in the tracks.

7. CONCLUSION

In this research, we proved that genomic data can often be re-identified in a distributed health environment. We developed and evaluated several algorithms, collectively termed RE-Identification of Data in Trails (REIDIT), that re-identify by using unique features in the sets of locations that patients visit. The REIDIT algorithms demonstrate that anonymity protection techniques neglecting to incorporate both computational and healthcare factors can be susceptible to re-identification. Moreover, the development of our models in a computational manner shifts the problem of anonymity analysis from *ad hoc* methods into a formal model. In the future, to evaluate anonymity protocols it necessary that researchers attack the problem with context dependent aspects in mind. Privacy protection methods can be tested against the current array of re-identification techniques, such as trail re-identification, to certify anonymity and thereby guarantee patient privacy.

Acknowledgments

The authors wish to thank Rema Padman, Robert Murphy, and Victor Weedn for helpful discussions when this research was in its early stages. In addition, the authors wish to thank the members of the Data Privacy Laboratory for their support and provision of an intellectually stimulating environment: Ralph Gross, Yiheng Li, Sherice Livingston, Elaine Newton, and Ben Vernot. This work was supported in part by Data Privacy Laboratory at Carnegie Mellon University.

References

[1] Altman RB and Klien TE. Challenges for biomedical informatics and pharmacogenomics. Annu Rev Pharmaco and Toxicol 2002; 42: 113-133.

[2] Dugas M, Schoch C, Schnittger S, *et al*. Impact of integrating clinical and genetic information. In Silico Biol 2002; 2: 383-391.

[3] National Research Council. For the record: protecting electronic health information. National Research Council. Washington, DC; National Academy Press. 1997

[4] Rothstein MA, editor. Genetic secrets: protecting privacy and confidentiality in the genetic era. New Haven; Yale University Press. 1997.

[5] Robertson JA. Privacy issues in second stage genomics. Jurimetrics 1999 Fall; 40: 59-76.

[6] Gulcher JR, Kristjansson K, Gudbjartsson H, and Stefansson K. Protection of privacy by third-party encryption in genetic research. Eur J Hum Genet 2000; 8: 739-742.

[7] Department of Health and Human Services. 45 CFR (Code of Federal Regulations), Parts 160 – 164. Standards for privacy of individually identifiable health information, Final Rule. Federal Register 2002; 67(157): 53182-53273.

[8] Wylie JE and Mineau GP. Biomedical databases: protecting privacy and promoting research. Trends Biotechnol 2003; 21(3): 113-116.

[9] de Moor GJ, Claerhout B, de Meyer F. Privacy enhancing technologies: the key to secure

communication and management of clinical and genomic data. Meth Info Med 2003; 42: 148-153.
[10] Klein TE, Chang JT, Cho MK, *et al*. Integrating genotype and phenotype information: an overview of the PharmGKB project: Pharmacogenetics Research Network and Knowledge Base. Pharmacogenomics J 2001; 1(3): 167-70.
[11] Haas DW, Wilkinson GR, Kuritzkes DR, *et al*. A multi-investigator/institutional DNA bank for AIDS-related human genetic studies: AACTG Protocol A5128. HIV Clin Trials 2003; 4(5): 287-300.
[12] Winkelmann BR. Pharmacogenomics, genetic testing and ethnic variability: tackling the ethical questions. Pharmacogenomics 2003; 4(5): 531-535.
[13] Sweeney L. Uniqueness of simple demographics in the U.S. population. LIDAP-WP4. Laboratory for International Data Privacy, Carnegie Mellon University. 2000.
[14] Dreiseitl S, Vinterbo SA, Ohno-Machado L. Disambiguation data: extracting information from anonymized sources. Proc AMIA Symp 2001: pp. 144-148.
[15] Malin B and Sweeney L. Inferring genotype from clinical phenotype through a knowledge based algorithm. Pac Symp Biocomput 2002: pp. 41-52.
[16] Malin B and Sweeney L. Re-identification of DNA through an automated linkage process. Proc AMIA Symp 2001: pp. 423-427.
[17] National Association of Health Data Organizations. NAHDO Inventory of State-wide Hospital Discharge Data Activities. Falls Church; National Association of Health Data Organizations. May 2000.
[18] Sweeney L. Weaving technology and policy together to maintain confidentiality. Journal of Law Med Eth 1997; 25: 98-110.
[19] State of Illinois Health Care Cost Containment Council. Data release overview. Springfield; State of Illinois Health Care Cost Containment Council. March 1998.
[20] Malin B and Sweeney L. Determining the identifiability of DNA database entries. Proc AMIA Symp 2000: pp. 537-541.
[21] Gaudet D, Arsnault S, Belanger C, Hudson T, Perron P, Bernard M, and Hamet P. Procedure to protect confidentiality of familial data in community genetics and genomics research. Clin Genet 1999; 55: 259-264.
[22] Vaszer LT, Cho MK, Raffin TA. Privacy issues in personalized medicine. Pharmacogenomics 2003; 4(2): 107-112.
[24] Grannis SJ, Overhage JM, and McDonald CJ. Analysis of identifier performance using a deterministic linkage algorithm. Proc AMIA Symp 2002:305-309.
[23] Victor TW and Mera RM. Record linkage of healthcare insurance claims. Medinfo 2001; 10(Pt 2): 1409-1413.
[25] Blakely T and Salmond C. Probabilistic record linkage and a method to calculate the positive predictive value. Int J Epidemiol. 2002; 31(6):1246-52.